AFIT/GAP/ENP/96D-8

UNFOLDING THE HIGH ENERGY ELECTRON FLUX FROM CRRES FLUXMETER MEASUREMENTS

THESIS

Brian D. McKellar, Captain, USAF

AFIT/GAP/ENP/96D-8

DTIC QUALITY INSPECTED

Approved for public release; distribution unlimited

19970224 064

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the US Government.

AFIT/GAP/ENP/96D-8

UNFOLDING THE HIGH ENERGY ELECTRON FLUX
FROM CRRES FLUXMETER MEASUREMENTS

Brian D. McKellar, B.S.
Captain, USAF

Approved:

Kirk A Mathews	26 Nov 96
Dr. Kirk A. Mathews, Thesis Committee Chairman	Date
Wm. F. Bailey	26 Nov 96
Dr. William F. Bailey, Committee Member	Date
D L Coulliett	26 Nov 96
Dr. David L. Coulliette, Committee Member	Date
Jeffrey B. Martin	26 Nov 96
Dr. Jeffrey B. Martin, Committee Member	Date

AFIT/GAP/ENP/96D-8

UNFOLDING THE HIGH ENERGY ELECTRON FLUX FROM CRRES FLUXMETER MEASUREMENTS

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the Requirements for the
Degree of Master of Science in Applied Physics

Brian D. McKellar, B.S.
Captain, USAF

December 1996

Approved for public release, distribution unlimited

Acknowledgments

I am deeply indebted to my thesis advisor and mentor, Professor Kirk A. Mathews. His guidance and insight have been priceless throughout this research project. He never failed to make time for me as I progressed in my growth as a graduate student. Dr. Mathews was a coach, a cheerleader, a fellow researcher, and an inspiration. Without his assistance, I never could have accomplished the work I set out to do.

Mr. A. Robert Frederickson; from the Geophysics Directorate at the USAF Phillips Laboratory at Hanscom AFB, MA; was the sponsor for my research. His assistance with the initial stages of the literature review provided information critical to the success of my project. For Mr. Frederickson's help with the definition of the problem and quantification of the instrument response functions, I am grateful.

The successful completion of this task relied heavily on the support I received from my faithful companions in the trek through life, Bill and Ed. Their loyalty and quality of companionship are without equal. May the mice of their dreams always be plump, lethargic, and not very bright.

It was a blessing to work with my fellow graduate students, for their support and understanding made this thesis a team effort. In particular, I am especially thankful for the assistance given to me by Capt Brad L. Beatty.

I owe a special debt to my mother, Rosemary McKellar, for she is most responsible for making me the man I am today. Her adherence to duty and dedication to discipline is beyond reproach.

Brian D. McKellar

Table of Contents

Page

Acknowledgments ii

List of Figures v

List of Tables vii

Abstract viii

I. Introduction 1

Background 1
Motivation for the Research 3
Statement of the Problem 4
Scope of the Problem 6
Research Objectives 7
Overview of the Thesis 8

II. The Theory of Unfolding 9

Theoretical Relationships 10
Iterative Solutions for Linear Systems 17

III. The High Energy Electron Fluxmeter 19

General Characteristics of the Instrument 19
The Radiation Belt Environment 21
The First Calibration 25
The Second Calibration 30

IV. Methodology for the Research 36

Numerical Experiment Method 37
The Experiment Plan 42

V. Numerical Experiments: Results and Observations 44

The Three Sets of Response Functions 45
Approximating the First Calibration with a Set of Idealized Response Functions ... 49
Approximating the Second Calibration with a Set
of Idealized Response Functions 52
Approximating the Second Calibration with the First Calibration 55
Incorporating the Various Types of Error into the Unfolding Calculations 58

VI. Conclusions and Recommendations 80

Summary of Results 80
Inferences from the Observations 81
Conclusions 83
Recommendations 84

Appendix A: The Fortran 90 Code 85

Appendix B: A Case Study for Comparison 127

Bibliography 129

Vita 131

List of Figures

Figure Page

1. Location of various Department of Defense space assets 3

2. A cross-sectional view of the High Energy Electron Fluxmeter 20

3. The omni-directional flux of electrons within the earth's radiation belts 24

4. The response functions for the HEEF as derived in the first calibration 29

5. Response functions for the first calibration as calculated by the code 30

6. The new absolute response function obtained from the partial recalibration 34

7. Possible response functions (extrapolated from the second calibration) as calculated by the code 35

8. The set of idealized response functions 46

9. The discretization of the idealized response functions 47

10. The discretization of the response functions for the first calibration 47

11. The discretization of the response functions for the second calibration 48

12. The flux values for test case 1 50

13. The flux values for test case 2 51

14. The flux values for test case 3 53

15. The flux values for test case 4 54

16. The flux values for test case 5 56

17. The flux values for test case 6 58

18. The flux values for test case 7 59

19. The flux values for test case 8 61

20. The flux values for test case 9 63

21. The flux values for test case 10 64

22. Percentage high and low relative errors vs. error ratio 66

23. A statistical error analysis, by energy bin, for test case 10 67

24. A statistical error analysis, by trial spectra, for test case 10 68

25. The flux values for test case 11 69

26. The flux values for test case 12 71

27. A statistical error analysis, by energy bin, for test case 12 72

28. A statistical error analysis, by trial spectra, for test case 12 73

29. The flux values for test case 13 74

30. The flux values for test case 14 76

31. A statistical error analysis, by energy bin, for test case 14 77

32. A statistical error analysis, by trial spectra, for test case 14 78

List of Tables

Table Page

1. Values of Key Parameters for the First Calibration of the Fluxmeter 27-28

2. Values of Key Parameters for the Second Calibration of the Fluxmeter 33

3. Expected Counting Errors for $\varphi^E(E) \sim E^{-4}$ with Second Calibration 79

4. A Summary of the Fundamental Results 81

AFIT/GAP/ENP/96D-8

Abstract

The Combined Release and Radiation Effects Satellite (CRRES) was launched on 25 July 1990 to collect measurements in the earth's radiation belts. One instrument, the High Energy Electron Fluxmeter (HEEF), measured the flux of electrons in 10 channels with energies between 1 MeV and 10 MeV. The channel sensitivities, $R_i(E)$, have been calibrated and partially re-calibrated. We explore the errors introduced in unfolding the electron flux spectrum from the channel measurements and the propagation and growth of calibration and measurement errors. Using numerical experimentation, we fold the responses with known spectra to obtain simulated measurements, add random measurement and calibration errors, and unfold the spectra as 10-bin histograms which are compared with histograms of the original spectra. We observe that the shape (of the response functions) is the major factor in the growth of error in unfolding and in determining which type of error dominates the unfolding process. We conclude that successful unfolding of the electron flux is critically dependent upon the shape of the response functions. The re-calibration of the HEEF must be accurately completed if reliable unfolds of the high energy electron flux are to be obtained.

UNFOLDING THE HIGH ENERGY ELECTRON FLUX
FROM CRRES FLUXMETER MEASUREMENTS

I. Introduction

In 1958 James Van Allen discovered zones of trapped ions and electrons encircling the planet. The particles which comprise these radiation belts find their origins both in the solar wind which blows past the planet and in the earth's lower atmosphere. Once these particles are injected into the region of space found between one and ten earth radii out from the planet, they are trapped by the earth's magnetic field. This dynamic system of charged particles is one of the fundamental regions in the near-earth space environment, and within this region operate many Department of Defense (DOD) assets. This is an environment where critical parameters must be accurately measured, basic comprehensive theories developed, and detailed computer generated models produced in order to allow the maximum utilization of DOD space assets. Unfolding the high energy electron flux from the available data base is one of the many steps required to cultivate an understanding of this region of space.

Background

The Combined Release and Radiation Effects Satellite (CRRES) was launched on an Atlas-Centaur booster on 25 July 1990, and was the first major scientific mission to the Earth's outer radiation belts since NASA's Small Scientific Satellite was launched two decades earlier [14]. CRRES was placed into a highly elliptical orbit with a perigee of

350 km (1.056 earth radii) and an apogee of 33,584 km (6.331 earth radii). This allowed the satellite to collect data from large portions of the earth's ionosphere and magnetosphere. The mission had a specified duration of one year, but the goal was three years. CRRES collected data until 9 October 1991, when catastrophic electronic failure terminated the mission [7].

The satellite was designed to conduct three classes of experiments in the near-earth space environment. The first class initiated active chemical releases in the ionosphere and magnetosphere. The second set investigated the natural radiation environment found in the inner and outer radiation belts. The final set of experiments studied irregularities in the low altitude ionosphere [1].

One of the instruments carried by the CRRES for the investigation of the natural radiation environment was a High Energy Electron Fluxmeter (HEEF) designed to measure the flux of electrons with energies between 1 MeV and 10 MeV. These high energy electrons primarily inhabit the earth's radiation belts [8]. Measuring their energy distribution requires that the fluxes be unfolded from the signals generated by the instrument using the response functions derived from the fluxmeter's calibration. This is a complex process with several components. The characteristics of the measured fluxes will depend on the energies of the electrons which the instrument measures, the response of the instrument to the various electron energies, the errors introduced by the ill-conditioned nature of the unfolding computations, the errors in the calibration process, and the errors in the signal measurements [11].

This thesis studies the unfolding of these high energy electron fluxes, incorporating all of the above listed aspects. The goal of the research was to determine how all of these parameters impact the unfolding process. This leads to a qualitative comprehension of the HEEF measurements. Without this detailed understanding it is extremely difficult to accurately describe the flux of high energy electrons in the earth's radiation belts.

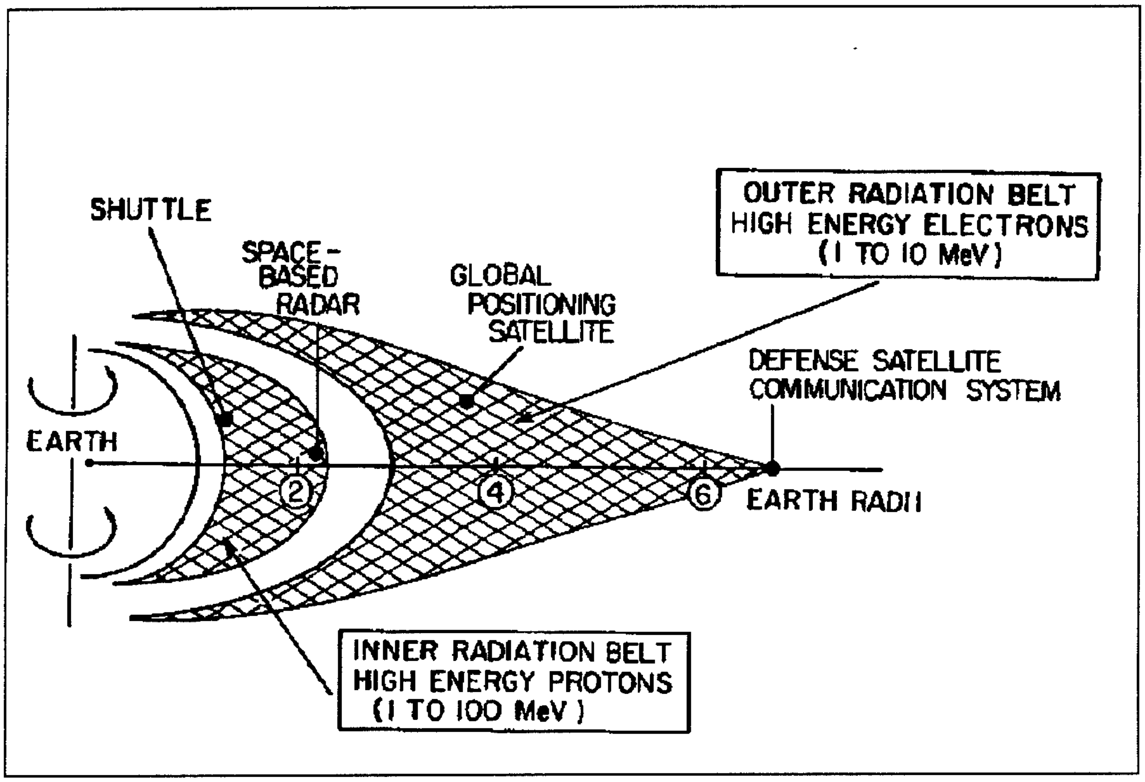
SHUTTLE
SPACE-BASED RADAR
GLOBAL POSITIONING SATELLITE
OUTER RADIATION BELT HIGH ENERGY ELECTRONS (1 TO 10 MeV)
DEFENSE SATELLITE COMMUNICATION SYSTEM
EARTH
2
4
6
EARTH RADII
INNER RADIATION BELT HIGH ENERGY PROTONS (1 TO 100 MeV)

Figure 1. Location of Various Department of Defense Space Assets [9]

Motivation for the Research

The region of space near the earth is neither empty nor benign, so it is important to develop an understanding of the dynamic nature of this operational arena. Assets critical to national security operate within the radiation belts; figure 1 shows where some of these systems are located. An excellent way to develop a working knowledge of this environment is to utilize computer models. These tools can analyze current conditions in the magnetosphere (this capability presently exists) and predict the future state of the magnetosphere (this capability is expected to occur by the year 2000). The motivation for this thesis grows from the desire to understand the interplay of the various components impacting the ability to make a measurement from a remote sensing platform located in space. The critical first step to modeling is ensuring the accuracy of the data used to initialize the model. In other words, to comprehend a dynamic system one must first be able to measure the elements found within that system. An integral component of this process is the ability to unfold energy fluxes from a given set of instrument signals and response functions. If the measurements are not sound, the model output will be suspect and the near-earth environment will not be accurately depicted. Sound measurements presume sound calibration and data reduction methodology.

A key input to interpreting the HEEF measurements (by unfolding the observed electron spectrum) is the sensitivity of the instrument channels to electrons, as a function of their energies. This sensitivity function is called (in the unfolding community and in this thesis) a *response function*. The response function for each measurement channel is itself measured when the instrument is calibrated. The HEEF was first calibrated before it was flown. Later, when the HEEF measurements were analyzed, this calibration was called into question. A second calibration (over part of the energy range of the HEEF) was performed on the backup instrument. The second set of response functions differed substantially from the sensitivities reported by the contractor after the first calibration.

Statement of the Problem

The research for this thesis initiated with the development of a working knowledge of the theory of numerical unfolding. Once a conceptual understanding was obtained, it could be applied to the HEEF carried on the CRRES. This, in turn, necessitates a detailed understanding of the design of the instrument. In particular, the instrument's sensitivities to electrons of various energy must be modeled analytically by a set of response functions obtained through a calibration. A comprehension of the morphology of the radiation belts completes the foundation of the knowledge required for the research.

To examine the impact of different parameters on numerical unfolding, one needs a tool which allows experimentation based on a well-founded methodology. The tool for this thesis is a computer, with the specific device being a program written in the language Fortran 90 and run on the Microsoft Fortran Power Station (v4.0) compiler. Hence, a large amount of the research effort was expended on producing a sound algorithm with which to study the impacts of all the various parameters involved with unfolding the high energy electron flux from the HEEF measurements. The capabilities of the code are closely linked with the methodology of the experiment.

A basic understanding of the research problem in conjunction with the proper tool allows a thorough experimental procedure to be followed. This provides a wealth of data for use in the analysis of the performance of the HEEF, and so output generation constitutes another step in the process. By considering the impact of different types of fluxes, different types of response functions, and different types of error, one can provide answers to a series of questions. First, can an idealized set of response functions be used to approximate unfolding with the first calibration of the instrument? Second, can an idealized set of response functions be used to approximate unfolding with the second calibration of the instrument? Third, can the first calibration be used as an approximation to the second calibration of the fluxmeter? Fourth, what magnitude of error is introduced by the unfolding technique used in the research. Finally, what is the magnitude of the error in the unfolded fluxes that is introduced from errors in the instrument calibration and errors in the signal measurements?

Laying a proper foundation of theoretical and instrumental knowledge in conjunction with the proper tool scales the scope of the research and produces a data set which aids in the characterization of this instrument. The various components of research undertaken during the course of this thesis were always measured against the primary goal, and that was analyzing the adequacy of the calibrations and the significance of the differences between them. The real performance of the HEEF can not be known until a definitive calibration has been done and is then evaluated in terms of its impact on uncertainties in unfolding spectra.

Scope of the Problem.

The research develops a basic understanding of unfolding theory and applies it to known flux measurements similar in nature to those made by the HEEF carried on the CRRES mission. An unfolding technique is tested with both idealized response functions and response functions generated from the calibrations of the instrument. This will quantify errors introduced by the unfolding technique. Uncertainty in the response functions and in the instrument measurements are incorporated into the analysis of the response functions. This determines which source of error will dominate the unfolding process, and it develops a confidence level for unfolded fluxes generated from the actual instrument calibrations. The culmination of the thesis will be the comparisons made between the two sets of calibrated response functions and their respective effects on the measurement of the high energy electron flux spectrum.

Research Objectives

The Air Force must understand the environment within which its space assets operate. Computer modeling is an excellent method of obtaining this comprehension. If models are to produce accurate output, the initial data set must contain as little error as possible. One requires a solid understanding of exactly what is being measured. The goals for this thesis center on the process of making these measurements and closely parallel the development of the experimental methodology.

The first objective of the research was developing an algorithm which incorporated pertinent aspects of unfolding theory and applying it towards producing unfolded fluxes similar to those believed to have been observed by the HEEF. This code, this first objective, would be the centerpiece of the research from which all the other objectives would be obtained. This algorithm would grow into the diagnostics package used to produce all of the results of the thesis. It is hoped this same package can be given to the sponsor at Phillips Laboratory and used for further diagnostic analysis. The second objective was to determine if a technique more robust than a trivial unfold was required for calculating unfolded fluxes. The third objective studied the differences between the two sets of response functions for the HEEF. The final objective quantified the impact of the various sources of error inherent to the unfolding process. The goal for this research was to characterize the performance of the HEEF response functions and to evaluate all of the key parameters which impact the unfolding problem. The objectives of the thesis all build upon the same foundation, and they all contribute to this one goal.

Overview of the Thesis

Chapter two explains the theory of numerical unfolding and how it is utilized by the Fortran 90 code. Chapter three details the characteristics of the fluxmeter. This includes explanations of the two calibrations performed on the instrument. Chapter four describes the methodology used to produce the test cases examined in the data and analysis section of the thesis. Chapter five lists and explains the test cases generated by the code. This includes the differences between idealized and actual instrument response functions and the impact of the different types of error on the unfolding technique. The last chapter states the conclusions reached during the research period.

II. The Theory of Unfolding

The process of unfolding can be very complex, and the form of the unfolding algorithm depends heavily upon the application to which it is applied. Unfolding schemes range from those which are rather unsophisticated (such as the technique employed in this research) to those which are very intricate (but generally require *a priori* information about the spectrum which is to be unfolded). This discussion of the theory starts with a quote which defines unfolding without using mathematical terminology:

> "The process of unfolding consists of finding, through some scheme or another, an unfolded spectrum which is physically reasonable and which produces unfolded signals which agree with the measurements to a degree justified by their accuracy. Errors arise in this process from three sources: measurement error in the detector, calibration error in the instrument, and the uncertainty which results from the ill-posed nature of the problem. [12]"

The goal of this chapter is to develop a set of mathematical relations which provide an in-depth description of the above shown quote. These relations, in turn, must be used to unfold fluxes from a given set of signals and response functions. This chapter also explains the inversion technique used by the Fortran 90 code to implement the unfolding scheme.

The process of unfolding is ill-posed because a continuous energy flux is represented by a limited number of data points, and in the case of this specific problem only ten points are available for constructing a flux. This limitation on the amount of available data can be the most severe constraint placed upon the unfolding problem. The process is also ill-posed because wide bin energy boundaries, dictated by the response functions, are introduced into the unfold. Unfortunately, ill-conditioning is also a concern. Any physically realistic instrument will have response functions which overlap in their respective energy ranges, and this means any calibration or measurement error introduced into the unfold may be greatly amplified [10]. There are cases in the research which clearly show the extent of this amplification. Even though the definition of unfolding is straightforward, producing unfolded fluxes which contain any degree of accuracy can be extremely difficult.

Theoretical Relationships

The cornerstone of the research methodology is a thought experiment, and the flow of this process is based on the theoretical relations showing the calculation of the instrument signals and the unfolded fluxes. The development shown below is based on a review of the literature [10],[11], [12], and [15].

The process begins by selecting a known exact flux, $\varphi^E(E)$, in conjunction with a set of response functions, $R_i^E(E)$, generated from a calibration which contains no error. These inputs are used to construct a set of exact instrument signals, y_i^E, in accordance with the following relationship:

$$y_i^E \equiv \int_0^\infty R_i^E(E) \cdot \varphi^E(E) \cdot dE \tag{2-1}$$

where *i* indexes the HEEF channels. The relation shown above is valid if the fluxmeter supplies a continuous set of measurements. Since no instrument provides a set of infinite signals, a discrete approximation for a continuous flux must be developed. In this research the continuous flux is approximated by a set of histograms which are defined in the following relation:

$$\varphi^E(E) \approx \sum_k \tilde{\varphi}_k \cdot \chi_k(E) \equiv \tilde{\varphi}(E), \tag{2-2}$$

where

$$\chi_k(E) = \begin{cases} 0 & E < E_{k-1} \\ 1 & E_{k-1} \leq E \leq E_k \ or \, . \\ 0 & E_k < E \end{cases} \tag{2-3}$$

and k indexes the wide bins into which the flux is unfolded. The goal is to find expansion coefficients, $\tilde{\varphi}_k$, such that $\tilde{\varphi}(E)$ closely approximates the exact flux, $\varphi^E(E)$.

Note that the basis functions, $\chi_k(E)$, for the approximation in equation (2-2) are orthogonal, because

$$\int_0^\infty \chi_j(E) \cdot \chi_k(E) \cdot dE = \delta_{jk} \cdot \Delta E_k \, , \tag{2-4}$$

where δ_{jk} is the Kronecker delta, and $\Delta E_k \equiv E_k - E_{k-1}$. (The set of energies, $\{E_o, E_1, \ldots, E_{10}\}$ partitions the energy range of interest into 10 *wide bins*.) Thus, the best approximation can be obtained by determining the expansion coefficients as follows. Multiply equation (2-2), with *j* replacing *k* as the dummy index of summation, by $\chi_k(E)$ and integrate, to obtain:

$$\int_0^\infty \chi_k(E) \cdot \varphi^E(E) \cdot dE = \int_0^\infty \chi_k(E) \cdot \sum_j \tilde{\varphi}_j \cdot \chi_j(E) \cdot dE \, . \tag{2-5}$$

Using the orthogonality of the basis functions, this simplifies to

$$\int_0^\infty \chi_k(E)\cdot\sum_j \tilde{\varphi}_j\cdot\chi_j(E)=\sum_j \tilde{\varphi}_j\cdot\int_0^\infty \chi_j(E)\cdot\chi_k(E)\cdot dE \tag{2-6}$$

$$=\sum_j \tilde{\varphi}_j\cdot\delta_{jk}\cdot\Delta E_k=\tilde{\varphi}_k\cdot\Delta E_k \tag{2-7}$$

Thus, the best coefficients are

$$\tilde{\varphi}_k=\varphi_k^E\equiv\int_0^\infty \chi_k(E)\cdot\varphi^E(E)\cdot\frac{dE}{\Delta E_k}, \tag{2-8}$$

where we denote this choice of expansion coefficients as φ_k^E, the *exact* coefficients. This last result also shows how to interpret these coefficients: φ_k^E is the average of the exact flux, as averaged over the k^{th} wide bin. With this choice of coefficients, we have the discretized exact flux,

$$\tilde{\varphi}^E(E)\equiv\sum_k \varphi_k^E\cdot\chi_k(E). \tag{2-9}$$

Now our objective in unfolding is to try to approximate the exact coefficients using the measurement data to construct *unfolded* coefficients, φ_k^U. In actual practice, we wouldn't know $\varphi^E(E)$, but we would have the measurements, y_i^M. We could unfold those data to find the values, φ_k^U, but we couldn't know how well the unfolded flux approximates the (unknown) exact flux. So for our simulation tests, we start with a postulated $\varphi^E(E)$, construct exact signals, optionally simulate measurement and/or calibration error, unfold to get the coefficients φ_k^U, and compare them to φ_k^E to see how well the unfold worked. Thus, the first step (after choosing $\varphi^E(E)$) is to numerically approximate the integrals in equation (2-1) by composite midpoint numerical quadrature:

$$y_i^E \approx \sum_j R_i\left(\frac{E_{j-1}+E_j}{2}\right)\cdot \varphi\left(\frac{E_{j-1}+E_j}{2}\right)\cdot \Delta E_j \tag{2-10}$$

where $\Delta E_j = E_j - E_{j-1}$ are small enough increments of energy that the sum approximates the integral with negligible error. (We refer to these increments as the *narrow bins*.)

Once a set of instrument signals has been calculated, random noise can be added to simulate error in the measurements. This is shown in the following relationship:

$$y_i^M = y_i^E + \delta y_i^{COUNT} . \tag{2-11}$$

It is important to note that random noise does not have to be added to the calculation, the measured signals (denoted by *M*) can be set equal to the exact signals and used for the unfolding process. Calibration error in the fluxmeter's response functions (or instrument sensitivity) can also be simulated in the numerical experiments:

$$R_k^C(E) = R_k^E(E) + \delta R_k^{CAL}(E) . \tag{2-12}$$

Once again, random noise does not have to be included. The calibrated response functions (denoted by *C*) can be set equal to the exact response functions.

Now we need equations to solve for the unfolded flux coefficients, φ_k^U. Substitute the unfolded flux, in the form

$$\varphi^U(E) = \sum_k \varphi_k^U \cdot \chi_k(E), \tag{2-13}$$

into equation (2-1), in the place of $\varphi^E(E)$; substitute the calibrated response functions, $R_i^C(E)$, for the exact response functions; and substitute the (simulated) measurements, y_i^M, for the exact measurements. This yields

$$y_i^M = \int_0^\infty R_i^C(E) \cdot \sum_k \varphi_k^U \cdot \chi_k(E) \cdot dE, \tag{2-14}$$

which can be rearranged as

$$y_i^M = \sum_k \left[\int_0^\infty R_i^C(E) \cdot \chi_k(E) \cdot dE \right] \cdot \varphi_k^U \ . \tag{2-15}$$

Since we know $R_i^C(E)$ and $\chi_k(E)$, we can approximate the response matrix elements, R_{ik}^C, using numerical quadrature (narrow bins):

$$R_{ik}^C \equiv \int_0^\infty R_i^C(E) \cdot \chi_k(E) \cdot dE \tag{2-16}$$

$$\approx \sum_j R_i^C\left(\frac{E_{j-1} + E_j}{2}\right) \cdot \chi_k\left(\frac{E_{j-1} + E_j}{2}\right) \cdot \Delta E_j \,. \tag{2-17}$$

Thus, we have the linear system of equations,

$$y_i^M = \sum_k R_{ik}^C \cdot \varphi_k^U \ , \tag{2-18}$$

to solve for the unfolded (wide bin average) fluxes, φ_k^U. This can be written in matrix notation as

$$\mathsf{R}^\mathsf{C} \cdot \vec{\varphi}^\mathsf{U} = \vec{\mathsf{y}}^\mathsf{M} . \tag{2-19}$$

If we choose to use more wide bins than there are instrument channels (measurements), then the system is underdetermined: the solution is not unique and *a priori* information is needed to select a solution from among the possible ones. In the case of the HEEF measurements, the only *a priori* information is that the fluxes should be nonnegative [12].

If we choose to use fewer wide bins than measurements, the system is over-determined: there is (typically) no solution, but a least squares approximate solution is uniquely determined. This provides some data averaging, but also a loss of resolution. With only 10 measurements, we choose to use 10 bins in the unfold, so the linear system should have a unique solution.

Once the unfolded flux is calculated, a comparison can be made with the exact wide bin average fluxes, φ_k^E. Incorporation of calibration error, counting error, discretization error, and error inherent to the unfolding process provides a set of detailed comparisons between the unfolded fluxes and the exact wide bin average fluxes. This set of calculations is the foundation for the characterization of the High Energy Electron Fluxmeter. It is worthwhile to list the four relations showing the various types of error:

- counting error:

$$y_i^M = y_i^E + \delta y_i^{COUNT}, \tag{2-20}$$

- error in the calibration:

$$R_k^C(E) = R_k^E(E) + \delta R_k^{CAL}(E), \tag{2-21}$$

- the resulting error in the unfolded (bin average) fluxes:

$$\varphi_k^U = \varphi_k^E + \delta\varphi_k{}^{UNFOLD}, \tag{2-22}$$

- and discretization error:

$$\tilde{\varphi}^E(E) = \varphi^E(E) + \delta\varphi(E)^{DISCRETIZATION}. \tag{2-23}$$

While it may be possible to reduce the discretization error by choosing basis functions that better approximate the variation of the exact flux within the bins, this requires either *a priori* information or an iterative unfold (or both) [11, 12]. Doing so has a more serious drawback: the expansion coefficients become hard to interpret. We chose to use the basis functions, $\chi_k(E)$, so that φ_k^E is the k^{th} bin average of the exact flux. Any other choice destroys this interpretation. The discretization error is unfortunate, but the best way to reduce it is to use a higher-resolution fluxmeter.

This development is incorporated into an experimental methodology that begins with generating scenarios that use idealized unfolds. This step demonstrates the poor results gathered from approximating the calibrated instrument responses with an idealized response. No error from the signal measurements or from the calibration is incorporated into these first calculations. The next step produces scenarios which show the differences between the two calibrations of the instrument. This displays the importance of the shape of the fluxmeter's response functions, for one calibration cannot be used as an approximation for the other. No error from the signal measurements or from the calibration is incorporated into this set of calculations. In the last step, calibration error and counting error are simulated. This shows the impact of the various types of error on the calculated fluxes and provides evidence of the need for a definitive recalibration of the HEEF.

Iterative Solutions for Linear Systems

The tool used to conduct the research for this thesis is a code that collects inputs from the user, produces a set of signals and responses, and solves equation (2-18) with an iterative technique. There are a variety of methods which could be used to solve the linear system, equation (2-19). The matrices and vectors involved with this application are not large, so the individual calculations do not necessitate major investments of computing time. As a result, efficiency was not a pressing requirement for selection of an iterative technique. Instead, stability of performance and ease of coding drove the selection process. The algorithm used by the Fortran 90 code is Jacobi iteration.

The Jacobi method iterates to solve a linear system of the form

$$\mathsf{A} \cdot \vec{\mathsf{x}} = \vec{\mathsf{b}}. \tag{2-24}$$

Decompose A into the sum of diagonal, D, lower triangular, L, and upper triangular, U, matrices: $\mathsf{A} = \mathsf{D} + (\mathsf{L} + \mathsf{U})$. Thus,

$$\mathsf{D} \cdot \vec{\mathsf{x}} = \vec{\mathsf{b}} - (\mathsf{L} + \mathsf{U})\vec{\mathsf{x}}. \tag{2-25}$$

The diagonal matrix, D, is trivially inverted, to get the fixed point relation,

$$\vec{\mathsf{x}} = \mathsf{D}^{-1}\vec{\mathsf{b}} - \mathsf{D}^{-1}(\mathsf{L} + \mathsf{U})\vec{\mathsf{x}}, \tag{2-26}$$

which is iterated, in the form

$$\vec{\mathsf{x}}^{i+1} = \mathsf{T}\vec{\mathsf{x}}^{i} + \vec{\mathsf{c}}, \tag{2-27}$$

where

$$\mathsf{T} = -\mathsf{D}^{-1}(\mathsf{L} + \mathsf{U}) \tag{2-28}$$

and

$$\vec{c} = D^{-1}\vec{b}, \qquad (2\text{-}29)$$

until converged. In this application, $A_{jk} = R^{C}_{jk}$, $x_k = \varphi^{U}_{k}$, and $b_j = y^{M}_{i}$ [2].

The Jacobi method requires an initial estimate of the flux, $\varphi^{initial}$, the subroutine uses $D^{-1} \cdot \vec{y}^{M}$. Iterations continue until the difference between the successive values of the unfolded flux are less than a relative tolerance of 1×10^{-8}. When this tolerance condition is reached, the subroutine passes the vector of unfolded flux coefficients back to the main program.

III. The High Energy Electron Fluxmeter

The High Energy Electron Fluxmeter (HEEF) was one of more than forty instruments carried by CRRES to study the earth's inner and outer radiation belts. It was designed to measure the flux of electrons with energies between 1 MeV and 10 MeV. In addition, the instrument had to be insensitive to protons with energies of up to several hundred MeV. The HEEF, which carries the alpha-numeric designator AFGL 701-4, collected a vital set of data, for these high energy electrons deposit a major portion of the radiation dose received by micro-electronic components found on satellites that operate within the radiation belts [1]. This data set was to be utilized in the construction of models that would predict various parameters in the belt region. Correctly unfolding the electron flux from this instrument's measurements is crucial to this modeling effort [9].

General Characteristics of the Instrument

The HEEF was built for the US Air Force by Panametrics under contract number F19628-87-C-0169. It detected electrons by using four active elements housed within the instrument casing. A cross-sectional view of the fluxmeter is shown in figure 2. The HEEF is shielded by tungsten collimators, so the electrons which are actually detected travel down the bore of the instrument as the satellite moves through space. Covering this bore opening is a 0.006" beryllium foil which prevents low energy electrons and protons from being detected. (The energy cutoff is 0.14 MeV for electrons and 1.3 MeV for protons.) Of the four active elements, two are solid state silicon surface barrier detectors (SSD's) which have a thickness of 700 μm. These detectors are thin enough to allow passage of 1 MeV electrons, but not so thin as to allow 10 MeV electrons to pass through the scintillator without depositing all of their energy. The third active element is the bismuth germanate (BGO) scintillator where the high energy electrons are actually counted. The shape of the energy pulse recorded by the two SSDs and the BGO scintillator allow discrimination between electrons and protons and also between the various electron energies. The fourth and final active element is a plastic scintillator anti-coincidence shield. This anti-coincidence shield, in conjunction with the collimators, serves to limit the error introduced by counting off-axis electrons. The instrument

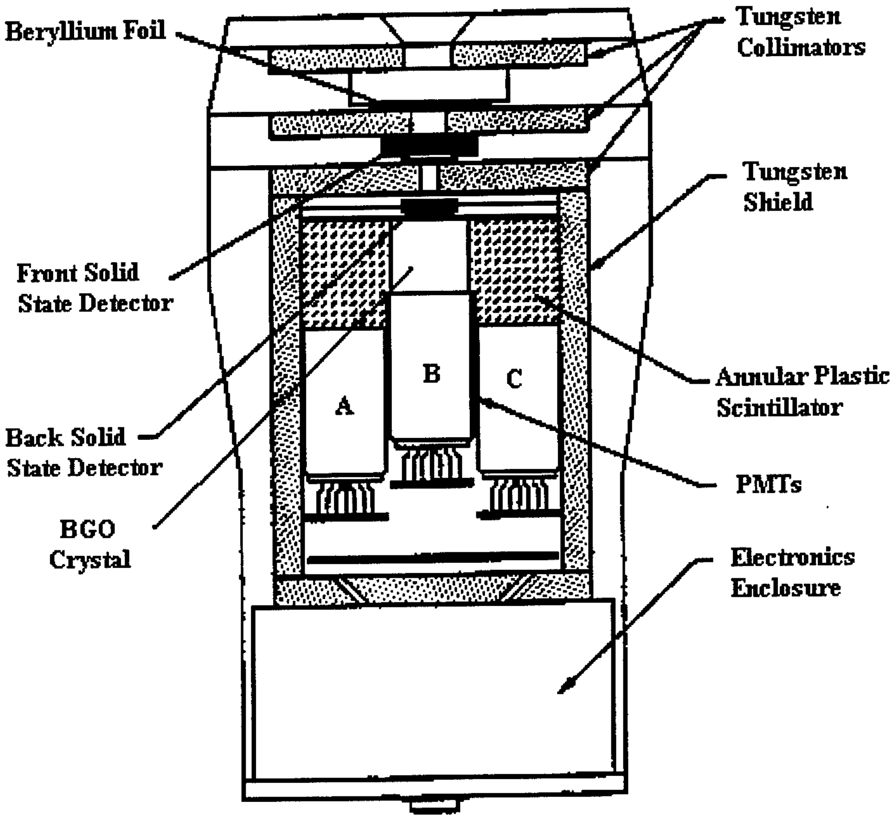

Figure 2. A cross-sectional view of the High Energy Electron Fluxmeter [4].

housing is made from magnesium so that bremstrahlung radiation will be minimized. In addition to studying the design of the HEEF, it is important to have a basic understanding of the radiation belts, and so it is necessary to discuss the environment within which this instrument operates.

The Radiation Belt Environment

High energy electrons are found in the earth's inner and outer radiation belts. This places the particles one to six earth radii above the surface of the planet. Figure 3 shows the omni-directional flux of electrons in the near earth environment [8]. Because these electrons are found in the magnetosphere (and not the ionosphere, where collisions between particles cannot be neglected), their dynamics can be characterized by three magnetic adiabatic invariants. To define an invariant, consider the motion of a particle described by a pair of variables (p_i, q_i) that are generalized momenta and coordinates. For each coordinate q_i that is periodic, the action integral J_i is defined as

$$J_i \equiv \int p_i \cdot dq_i \tag{3-1}$$

If the above relation is integrated over a complete period of q_i, with specified initial conditions, it is considered an invariant. This action integral remains invariant even if a property of the system changes, but this change must be slow (an adiabatic change) as compared to relevant periods of the system. In addition, the change in the system cannot be related to the period. If the electron is in an initial state of motion and then undergoes an adiabatic change in some system property, it will be in a new state of motion. The final state of motion is such that the value of its action integral will be the same as the action integral corresponding to the initial state of motion. The magnetic adiabatic invariants for the magnetosphere are associated with the gyration motion around a given set of magnetic field lines, the longitudinal motion parallel to the magnetic field lines, and the drift motion perpendicular to the field lines [13].

The first magnetic invariant is based on the cyclotron motion of the electron, and its mathematical representation is

$$B \cdot \frac{d\mu}{dt} = 0, \tag{3-2}$$

where B is the magnetic field intensity and μ is the magnetic moment of the electron. The relation shown in equation (3-2) shows the magnetic moment is independent with respect to time (since the earth's magnetic field strength is not 0) and is a constant for the guiding center of motion. Since the magnetic moment of the electron is constant, the total magnetic flux enclosed by the cyclotron motion of the electron must also be constant. Note, however, that the perturbation time of the earth's magnetic field must be significantly longer than the cyclotron period.

The second magnetic invariant is based on the longitudinal motion parallel to the magnetic field lines, and its mathematical representation starts with the definition of the appropriate action integral:

$$J = \int m \cdot v_{par} \cdot ds\,, \tag{3-3}$$

where m is the mass of the electron, v_{par} is the component of velocity parallel to the earth's magnetic field lines, and ds is the differential path length. The invariant is

$$\frac{dJ}{dt} = 0, \tag{3-4}$$

This implies the electron will move (parallel to the earth's magnetic field lines) in such a way as to preserve the total length of the particle trajectory. This invariant holds if the perturbation time scale is longer than the time it takes the electron to travel between the mirror points found near each magnetic pole.

The third magnetic invariant is based on the drift motion perpendicular to the field lines, and the action integral is defined as

$$J = \int m \cdot W_{per} \cdot d\phi, \tag{3-5}$$

where W_{per} is the drift velocity perpendicular to the earth's magnetic field lines and ϕ is the azimuthal angle. As in the case for the second invariant, the definition for the third invariant is

$$\frac{dJ}{dt} = 0. \tag{3-6}$$

The guiding center drift motion conserves the total magnetic flux within its drift path. This invariant is conserved as long as the perturbation time scale is longer than the drift period of the electron.

These invariants provide a general description of electron motion within the inner and outer radiation belts. By considering this theoretical basis one gains insight into the types of fluxes which may be measured in this region of space. Texts which describe the magnetosphere often assume these electron distributions are Maxwellian [8], [13], and this provides one functional form for fluxes used in the numerical experimentation. The propagation and growth of signal and calibration errors with this spectrum should also be studied. It is important to note that the flux based on theory cannot be taken as infallible.

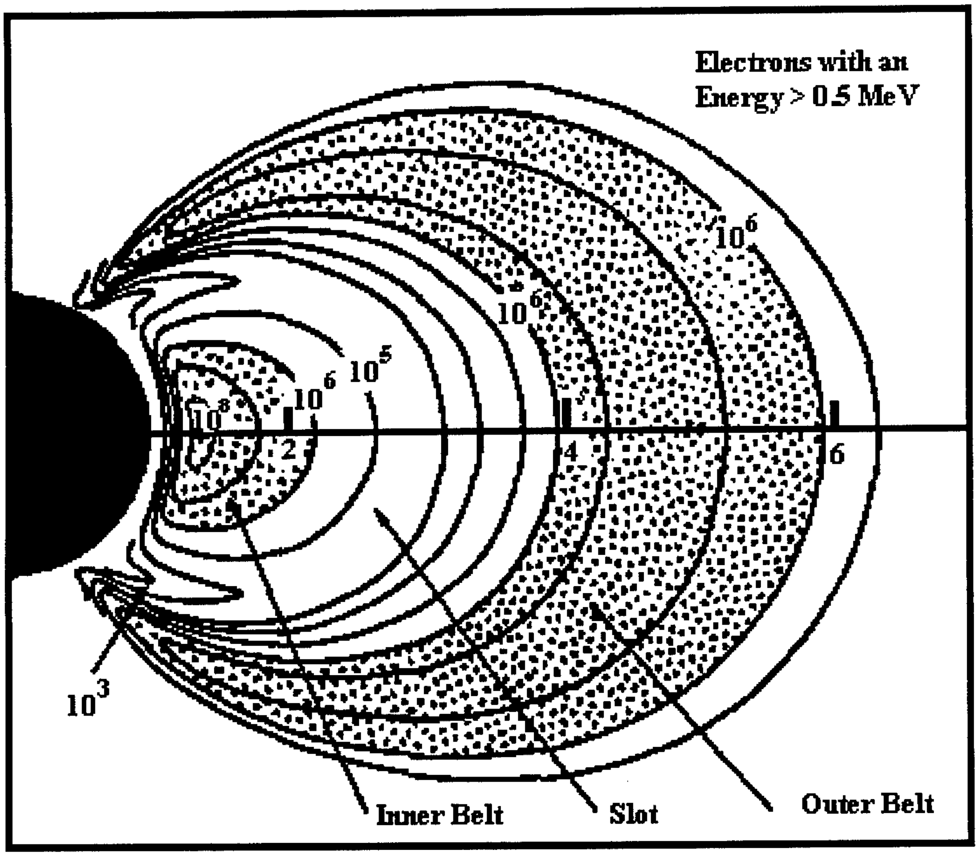

Figure 3. The omni-directional flux of electrons within the earth's radiation belts [8].

If it is assumed to be correct, there would be no point in making measurements in the first place. The data collected by CRRES, and other platforms like it, must validate the theory.

The First Calibration of the Instrument

The calibration of the HEEF is very important to the unfolding process, for it is this procedure which determines the sensitivity of the instrument. The value of each channel response will impact the number and energy of electrons in the unfolded flux. Perhaps more importantly, the shape of the response function will impact the growth and propagation of counting and calibration errors. If the instrument is not properly calibrated, it will be extremely difficult to produce an accurate unfolded flux.

The fluxmeter's first calibration was conducted at two different locations. The calibration over the energy range from 0.75 MeV to 1.75 MeV was performed at the Goddard Space Flight Center's Van de Graaff accelerator in July of 1987. The calibration covering the range of energies from 1.3 MeV to 10.8 MeV was done at the Rome Air Development Center's Linear Accelerator (no date given). These two calibration tests were used to derive the instrument's first set of response functions [3].

The calibration experiment is explained in brief so the reader will have a basic understanding of the technique. The two different tests produced values which were used to construct a geometric factor, defined as

$$G(E) = 2\pi \cdot \int_0^{\theta_{\max}} A(E,\theta) \cdot \sin(\theta) \cdot d\theta \tag{3-7}$$

where $A(E,\theta)$ is the effective detector area as a function of energy (E) and angle of incidence of the electrons (θ) and $\theta_{\max}$ is the largest angle for which $A(E,\theta)$ is non-vanishing. The calibration of the instrument produced values used to construct analytical expressions for the geometric factor. The contractor's report [3] gave the following expressions for the geometric factor:

$$G(E)=\begin{cases}\exp\left(-5.829\cdot E^2+21.452\cdot E-14.985\right) & 1.00\le E\le 1.75\ MeV\\ \exp\left(-0.378\cdot E^2+2.553\cdot E+1.373\right) & 1.75<E\le 2.8\ MeV\\ 700\cdot\left(1-\dfrac{1.75}{E-0.2}\right)^{1.2} & E>2.8\ MeV\end{cases},\quad (3\text{-}8)$$

where E is the energy of the electron. During the course of the research it was determined that the expression shown in equation (3-8), when plotted, contained a serious discontinuity. The response functions used in the Fortran 90 code (based upon the first calibration) incorporate a geometric factor given by

$$G(E)=\begin{cases}\exp\left(-5.829\cdot E^2+21.452\cdot E-14.985\right) & 1.00\le E\le 1.75\ MeV\\ \exp\left(-0.378\cdot E^2+2.553\cdot E+1.373\right) & 1.75<E\le 2.8\ MeV\\ 700\cdot\left(1-\dfrac{1.69}{E+0.2}\right)^{1.2} & E>2.8\ MeV\end{cases},\quad (3\text{-}9)$$

which eliminated the discontinuity.

In addition to the geometric factor, a relative response factor was measured during the calibration of the HEEF. The relative response is the probability that an electron with energy E will be counted in channel i. For channels 1 through 6, the analytical expression for this probability is:

$$R_i^{rel}(E)=R_i^{\max}\cdot\exp\left[-\left(E-E_i^{pk}\right)^2\Big/\left(2\sigma_i^2\right)\right],\qquad (3\text{-}10)$$

while for channels 7 through 10, it is

$$R_i^{rel}(E)=\begin{cases} R_i^{\max}\cdot\exp\left[-\left(E-\left(E_i^{pk}-\Delta E_i\right)\right)^2\big/\left(2\sigma_i^2\right)\right] & E<E_i^{pk}-\Delta E_i \\ R_i^{\max} & E_i^{pk}-\Delta E_i\le E\le E_i^{pk}+\Delta E_i \\ R_i^{\max}\cdot\exp\left[-\left(E-\left(E_i^{pk}+\Delta E_i\right)\right)^2\big/\left(2\sigma_i^2\right)\right] & E>E_i^{pk}+\Delta E_i \end{cases}. \quad (3\text{-}11)$$

The values for the variables in the above expressions (in addition to other key parameters for the first calibration) are found in Table 1 [3].

Table 1. Values of key parameters for the first calibration of the fluxmeter.

Instrument Channel	*i*	Bin Boundary Energy Value [MeV]	Bin Energy Width Value [MeV]	Energy at the Peak Bin Response, E_i^{pk} [MeV]	Value of the Peak Bin Response, $R_i^{\max}$ $[cm^2-sr]$
LL-L1	1	1.04 and 1.56	0.52	1.30	3.5 E-04
L1-L2	2	1.56 and 2.09	0.53	1.82	1.1 E-03
L2-L3	3	2.09 and 2.58	0.49	2.35	1.8 E-03
L3-L4	4	2.58 and 3.04	0.46	2.80	2.4 E-03
L4-L5	5	3.04 and 3.56	0.52	3.30	3.2 E-03
L5-L6	6	3.56 and 4.06	0.50	3.80	3.8 E-03
L6-L7	7	4.06 and 5.02	0.96	4.55	4.7 E-03
L7-L8	8	5.02 and 6.09	1.07	5.55	5.2 E-03
L8-L9	9	6.09 and 8.07	1.98	7.08	5.6 E-03
L9-L10	10	8.07 and 10.03	1.96	9.05	6.0 E-03

The response function for each channel (also referred to as the absolute response) of the HEEF is the product of the geometric factor and the relative response:

$$R_i(E)=G(E)\cdot R_i^{rel}(E)\cdot 10^{-5}. \quad (3\text{-}12)$$

where 10^{-5} is a factor used to scale the values of the absolute responses to those shown in reference 3. The source of this scaling factor appears to be a mistake in reference 3.

Figure 4 shows the response functions from the first calibration as shown in the calibration report [3]. Figure 5 shows the response functions for the first calibration as calculated by the Fortran 90 code.

The response functions calculated by the code closely match those provided by the first calibration. Inspection of figures 4 and 5 shows that, with the exception of the first three energy channels, little overlap of the successive channel responses is present. This decoupling of the instrument responses makes the unfolding process well conditioned. Unfortunately, the fluxes obtained from initial electron counts made by the HEEF cast doubt on this first calibration [4].

Table 1, con't. Values of key parameters for the first calibration of the fluxmeter.

Instrument Channel	i	$R_i^{\max}$	σ [MeV]	ΔE_i
LL-L1	1	0.919	0.234	-----
L1-L2	2	0.914	0.234	-----
L2-L3	3	0.925	0.234	-----
L3-L4	4	0.896	0.221	-----
L4-L5	5	0.886	0.234	-----
L5-L6	6	0.905	0.221	-----
L6-L7	7	0.997	0.293	0.15
L7-L8	8	0.997	0.340	0.15
L8-L9	9	1.000	0.357	0.58
L9-L10	10	1.000	0.425	0.50

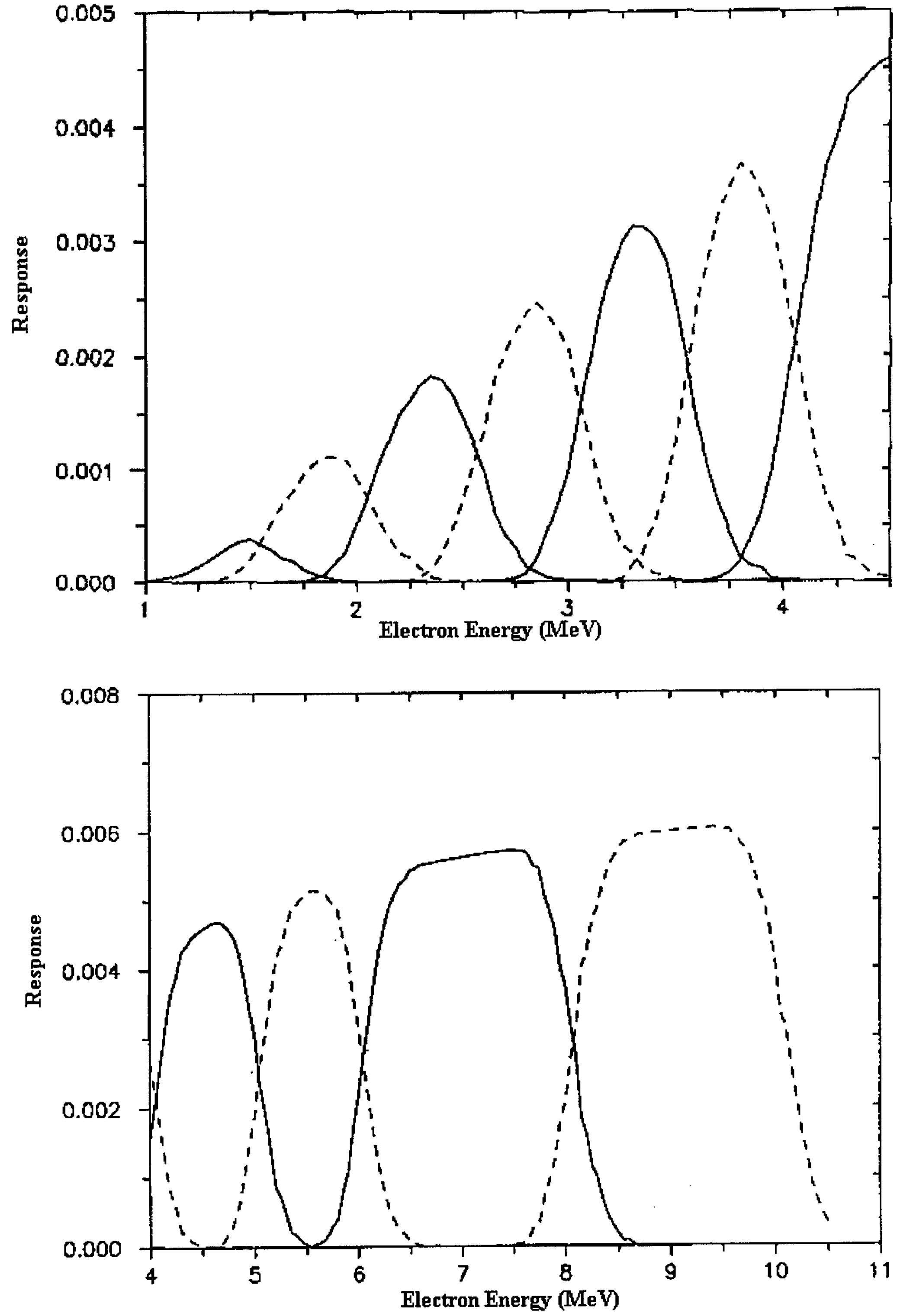

Figure 4. The response functions for the HEEF as derived in the first calibration [3].

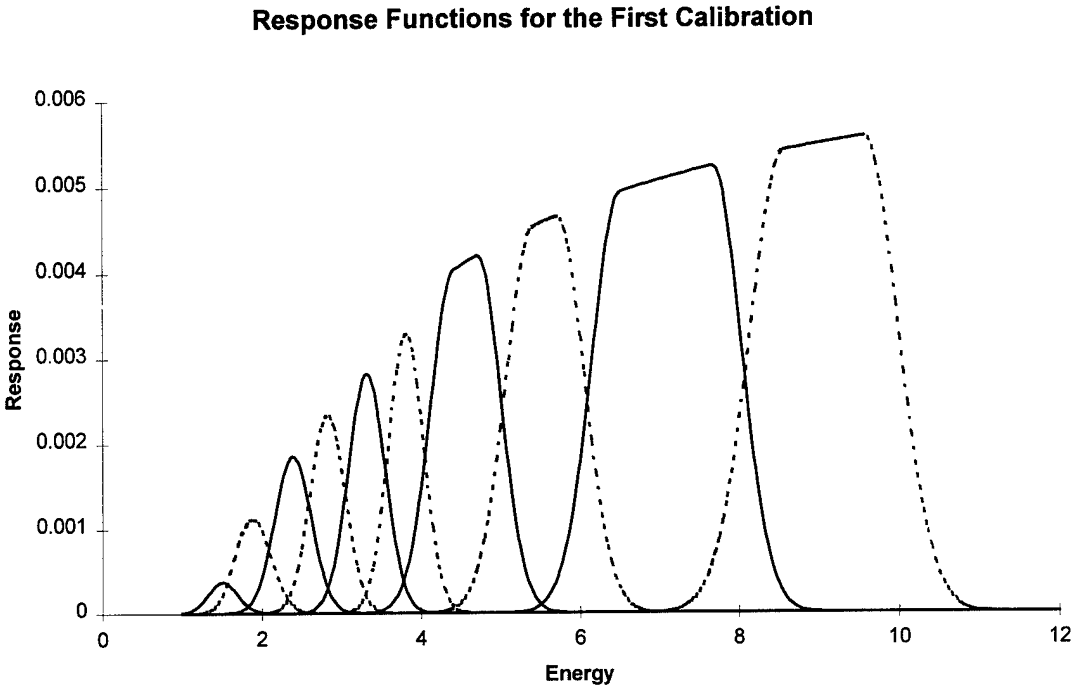

Figure 5. Response functions for the first calibration as calculated by the code.

The Second Calibration of the Instrument

As the initial spectrum measurements made by the CRRES were evaluated, it was discovered that the fluxmeter was operating in an environment with a temperature (of the actual instrument) different from the expected value. In addition, the voltage powering the instrument suite which contained the HEEF was at a different setting than that used for the calibration testing. As a consequence of these discoveries, minor corrections to the fluxmeter's response functions were made (reference 4 which describes this minor adjustment does not show a plot of the altered response functions, and there are no analytical representations listed) in order to obtain signal counts more in line with expected values. In spite of this minor correction to the instrument sensitivities, the electron fluxes derived from the HEEF signal measurements were still considered to be in error. It was decided to perform a partial recalibration of the HEEF [4].

The first High Energy Electron Fluxmeter had already been launched with the satellite, so it was not possible to re-calibrate the original instrument. What was possible was to conduct a recalibration on the spare HEEF designed and built for the CRRES. The contractor who built both fluxmeters claimed the designs were identical [6]. After reading the technical report which describes this recalibration, it appears that the second HEEF was not calibrated with the first instrument, so there is no common calibration test with which to compare the response functions of the two instruments. The fluxmeter carried on the satellite does contain an internal radioactive source with a low level of activity. This may be used for calibrating the instrument in space, but the bore must be closed to the natural radiation environment. The response functions for the two instruments may have been compared by using these internal sources, but this is not mentioned in any of the calibration reports.

The partial recalibration testing was conducted at the Massachusetts Institute of Technology's (MIT) Van de Graaff accelerator (no date given). This calibration covered the energy range from 0.3 MeV to 2.7 MeV. The procedures for calculating the absolute response of the instrument appear to be the same as those used in the first calibration, but this is difficult to ascertain from the technical report. No plots or analytical expressions for the relative responses are given. It appears that the primary difference in the new absolute response functions result from an estimated 20 % error in the determination of the geometric factor used in calculating the first set of response functions. The exact type of error is not made clear. The reference [6] does include a plot showing the new absolute response functions over the energy range from 0.3 MeV to 2.7 MeV, but no analytical expressions are given (which would allow the plot to be reconstructed).

Although table interpolations could have been used in the range in which the recalibration was performed, some form of extrapolation to higher energies was needed. The intent was to develop a new set of response functions (for all the channels) that would characterize the shape of the low-energy channels and apply it to the higher channels. A convenient, piecewise-defined, generic response function was devised. The low-energy segment is a Gaussian that rises up to the peak response, R_i^{pk}, at E_i^{pk}, where its slope is zero. The high energy segment is a constant, at some fraction, f_i^{tail}, of the peak sensitivity, for $E \geq E_i^{tail}$. These segments are joined by a cubic spline segment that matches the values and slopes (zero) of the tail segments where it joins them. Specifically, this assumed form is

$$R_i(E) = \begin{cases} \exp\left[-\left(E_i^{pk} - E\right)^2 \Big/ \left(2\sigma_i^2\right)\right] \cdot R_i^{pk} & E < E_i^{pk} \\ \left(1 + x^2 \cdot \left(1 - f_i^{tail}\right) \cdot (-3 + 2 \cdot x)\right) \cdot R_i^{pk} & E < E_i^{tail} \\ f_i^{tail} \cdot R_i^{pk} & E \geq E_i^{tail} \end{cases} \tag{3-13}$$

where E is the electron energy and

$$x = \left(\frac{E - E_i^{pk}}{E_i^{tail} - E_i^{pk}} \right). \tag{3-14}$$

The parameters shown in equations (3-13) and (3-14) are listed in table 2. These values are taken from two different sources. Values that could be taken from the plot shown in

Table 2. Values of key parameters for the second calibration of the fluxmeter.

Instrument Channel	i	(E_{i-1}, E_i) [MeV]	ΔE_i [MeV]	E_i^{pk} [MeV]	E_i^{tail} [MeV]	R_i^{pk} $[cm^2 - sr]$	$R_i^{tail} = f_i^{tail} \cdot R_i^{pk}$ $[cm^2 - sr]$
LL-L1	1	1.15, 1.51	0.36	1.35	2.02	6.0 E-04	4.8 E-04
L1-L2	2	1.51, 1.85	0.34	1.70	2.73	1.0 E-03	5.2 E-04
L2-L3	3	1.85, 2.54	0.69	2.25	3.38	1.3 E-03	6.5 E-04
L3-L4	4	2.54, 3.03	0.49	2.80	4.20	2.4 E-03	1.2 E-03
L4-L5	5	3.03, 3.54	0.52	3.30	4.95	3.2 E-03	1.6 E-03
L5-L6	6	3.54, 4.21	0.67	3.80	5.70	3.8 E-03	1.9 E-03
L6-L7	7	4.21, 5.15	0.95	4.55	6.83	4.7 E-03	2.4 E-03
L7-L8	8	5.15, 6.66	1.51	5.55	8.33	5.2 E-03	2.6 E-03
L8-L9	9	6.66, 8.55	1.89	7.08	10.62	5.6 E-03	2.8 E-03
L9-L10	10	8.55, 10.05	1.48	9.05	13.58	6.0 E-03	3.0 E-03

the technical report for the partial recalibration [6] were used, and the other values were taken from information provided for the first calibration. Perhaps the most important finding of the second calibration is the change in shape of the response function. The

Table 2, con't. Values of key parameters for the second calibration of the fluxmeter.

Instrument Channel	i	σ [MeV]	$f_i^{tail} = R_i^{tail} / R_i^{pk}$
LL-L1	1	0.234	0.36
L1-L2	2	0.234	0.34
L2-L3	3	0.234	0.69
L3-L4	4	0.221	0.49
L4-L5	5	0.234	0.52
L5-L6	6	0.221	0.67
L6-L7	7	0.293	0.95
L7-L8	8	0.340	1.51
L8-L9	9	0.357	1.89
L9-L10	10	0.425	1.48

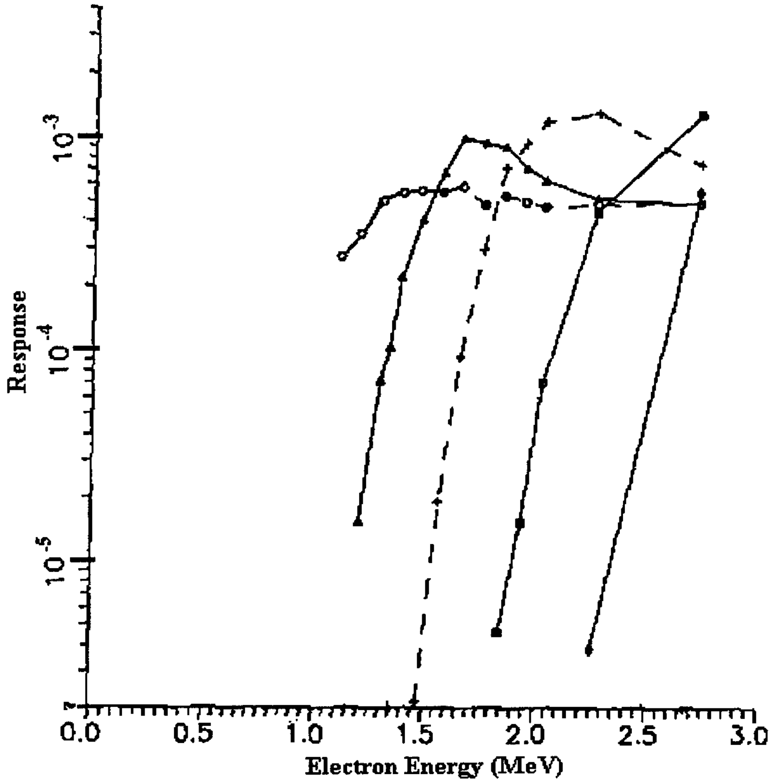

Figure 6. The new absolute response function obtained from the partial recalibration [6].

addition of a high energy tail seriously impacts the performance of the unfolding procedure and the quality of the unfolded fluxes which it produces. Figure 7, from reference 6, shows the absolute response functions derived from the partial recalibration of the HEEF. Figure 8 shows a plot of the response functions for the recalibration as calculated by the Fortran 90 code.

The two sets of response functions shown in figures 5 and 8, together with the four idealized responses used by the code, constitute one half of the requirement for unfolding the electron flux. The other requirements are the signal measurements made by the instrument. Characterizing the difference (in effect on the unfolded spectra) between these two response functions is a cornerstone of the research, and a significant portion of the results contained within the thesis pertain to this aspect of the problem.

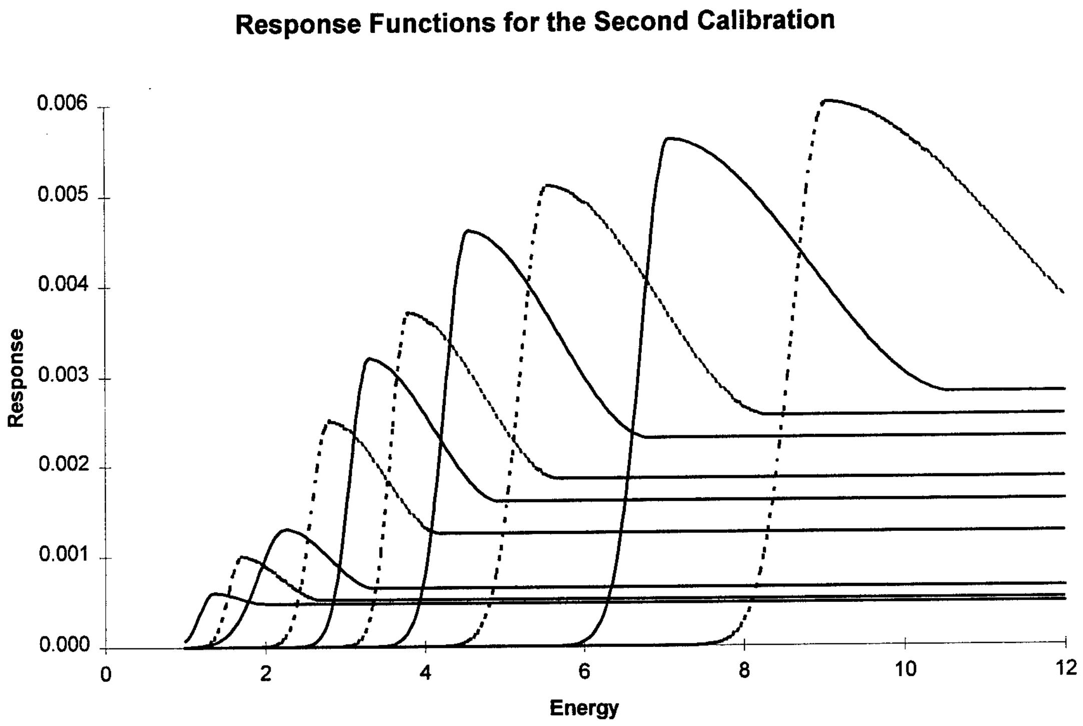

Figure 7. Possible response functions (extrapolated from the second calibration) as calculated by the code.

IV. Methodology for the Research

This thesis answers a series of questions posed by the project's sponsor in the Geophysics Directorate at the USAF Phillips Laboratory at Hanscom AFB, MA. These concerns all center on the performance of the High Energy Electron Fluxmeter. The characterization of this performance depends upon quantifying the effects of the differences between the two sets of calibrated instrument response functions. A sound experimental methodology must be employed if a successful performance characterization is to be accomplished, and this chapter explains that methodology.

A rational method for conducting numerical experiments produces answers to the questions which were posed in the first chapter. For instance, is there a need for anything beyond a trivial unfold? Can an idealized set of response functions be used as an approximation for instrument sensitivities quantified in the first and second calibrations of the instrument? Are the response functions from the first calibration a suitable substitute for the sensitivities developed from the partial recalibration? How do errors inherent to the unfolding technique alter the values of the unfolded electron fluxes? Is there a significant difference in the errors introduced through unfolding with the two different sets of response functions? Finally, how do the presence of errors in the instrument calibration and in the measured counts influence the value of the unfolded flux? In other words, is one type of error dominant over the others? Answers to these questions provide valuable input for the sponsor and his continuing efforts to refine the HEEF's performance [5]. The methodology for conducting this research must be complete, and so the experimental plan will be explained. In addition, the steps in a generic numerical experiment will be detailed. This provides the reader with an understanding of the approach the researchers took to solve this problem. Before discussing the experiment plan, the methodology for conducting a numerical experiment is presented.

Numerical Experiment Method

The purpose of a single experiment is to determine how well a particular spectrum could be unfolded if the actual response functions had a particular form and if a (possibly different) set of response functions were used in unfolding. The steps used to conduct such an experiment are as follows.

1. The experiment initiates with the selection of an exact spectrum, $\varphi^E(E)$. The experimenter may choose from seven different functional forms, of which the two most relevant to this research are

$$\varphi^E(E) = E^{-p} \tag{4-1}$$

where $p > 0$ is an input to the code, and

$$\varphi^E(E) = (1.128)/\tau \cdot (E/\tau)^{1/2} \cdot \exp\left(\frac{-E}{\tau}\right) \tag{4-2}$$

where τ, the temperature (in MeV) of the Maxwell-Boltzmann distribution, is also an input to the code. The actual electron spectrum present in the near-earth space environment is not known, but these test cases are designed as thought experiments and so the user can select the flux. Without knowing the exact spectrum, it would not be possible to quantify the differences between the response functions and the impacts of the various sources of error.

2. The next step is to fold the exact flux with a set of response functions in order to calculate a set of simulated instrument signals, y_i^E. This uses a fine mesh (referred to as narrow bins in the Fortran 90 code) discretization of the continuous functions in order to calculate the integrals with negligible quadrature error, using equation (2-10). Various sets of response functions are used in order to explore how the shape of the functions affects the accuracy of the unfolded flux. Chapter three contains mathematical definitions of the two sets of HEEF response functions.

3. At this point in the experimental process, the researcher has a set of exact instrument signals calculated by folding the exact flux with a set of response functions where it was assumed there was no error in the instrument calibration. The experimenter may now add simulated counting error, δy_i^{COUNT}, to the exact instrument signals or add calibration error, $\delta R_k^{CAL}(E)$, to the response functions. The exact signals, y_i^E, are positive. The standard deviation of a Poisson-distributed count, *n*, is $\sigma^{count} = \sqrt{n}$. The user selects the relative uncertainty, ε^{count}, in the counts for the channel with the maximum signal. (In practice, the maximum count was often $\sim 10^4$, so that $\varepsilon^{count} \sim 1\%$.) The response functions used here can be considered to be in arbitrary units. In any case the actual counts depend upon the duration of the count and the intensity of the flux. Therefore, the relative uncertainty entered in the code is scaled inversely with the square

root of the signal. (A lower signal, compared to the maximum signal, implies a smaller count and greater relative uncertainty.) Thus:

$$\delta y_i^{COUNT} = \left(\frac{\varepsilon^{count}}{100} \cdot y_i \right) \cdot \xi_i^G \cdot \left(\frac{y_{Max}}{y_i} \right)^{1/2} \tag{4-3}$$

Note that ε^{count} is entered as a percentage, whence the divisor, 100. The factor ξ_i^G is a random sample from a standard normal (Gaussian) distribution, i.e., with zero mean and unit variance. Although electron counts have Poisson statistics, the normal distribution is an adequate approximation, presuming each channel has about 10 or more counts. This is an important consideration if the Fortran code is used for numerical experiments involving fluxes that decrease rapidly with increasing electron energy values.

The calibration error is modeled as

$$\delta R_i^{CAL}(E) = \left(\varepsilon^{CAL} \cdot R_i^E(E) \right) \cdot \xi_i^U \tag{4-4}$$

where $R_i^E(E)$ comprises the set of exact response functions, ε^{CAL} is an error percentage input by the user, and the factor ξ_i^U is a random sample from a uniform distribution, with $-1 < \xi_i^U < 1$. This error formulation simulates systematic error in the calibration of the instrument channel (with a different random error for each channel). The Fortran 90 code has an option which calculates an energy dependent calibration error, but that was not used for the numerical experiments. (It was not needed in the testing reported here, but it remains a part of the code.) If the researcher does not wish simulate errors, ε^{count} and/or ε^{CAL} can be set equal to 0. In either case (with or without error), the user now has a set of simulated measured signals, y_i^M, and a set of simulated calibrated response functions,

$R_i^C(E)$, to use for unfolding the electron flux. These two quantities are passed to a subroutine in the Fortran code which solves for the unfolded flux.

4. In this step, the histogram panels, $\chi_k(E)$, (referred to as wide bins within the computer code) are constructed as basis functions for unfolding. These histogram panels partition the electron energy range of interest (as implemented, 1 to 12 MeV). In addition, these panels approximate the shape of the response functions (or the difference between adjacent ones), and thus, the bin boundaries are at the lower threshold of the responses. The highest bin boundary is either selected to ensure complete coverage of the energy range of interest or it is selected based upon input from the calibration reports. This results in wide bin energy boundaries at $E_0, ..., E_{10}$. The values for these bin boundaries are listed in the tables found in chapter three.

5. Now the computer code unfolds the measured signals to get the expansion coefficients, φ_k^U which (we hope) are good approximations to the bin-average exact fluxes, φ_k^E, as discussed in chapter 2.

6. In the last step, the performance of the unfold with respect to the selected set of response functions is evaluated. This is done graphically or with statistical measures of performance. Graphically, plots of $\tilde{\varphi}^E(E)$ and $\varphi^U(E)$, which are histograms, are overlaid with a (smooth) plot of $\varphi^E(E)$.

When random counting and/or calibration errors are simulated, the resulting unfolded flux (set of values, φ_k^U) depends on the specific random numbers that are drawn in sampling from the appropriate distribution. To characterize the effect of these errors, an ensemble of simulations, differing only in the seeding of the random number generator, is run. Then, we examine various statistics that are norms of the errors. We use the *error ratio* as a measure of the relative difference between a benchmark value, such as φ_k^E, and an unfolded approximation to it, such as φ_k^U:

$$\varepsilon_{rel}\left(\varphi_k^U, \varphi_k^E\right) = \frac{\varphi_k^U - \varphi_k^E}{\left(\left|\varphi_k^U\right| + \left|\varphi_k^E\right|\right)/2} . \tag{4-5}$$

Note that, if $\varphi_k^U \approx \varphi_k^E$, then $\varepsilon_{rel}\left(\varphi_k^U, \varphi_k^E\right)$ is approximately the conventional relative error (expressed as a fraction, rather than as a percentage). Where L simulations are run ($\ell = 1, \ldots, L$), we can use an average or maximum (over ℓ or k or both) of the absolute values of the error ratios to characterize performance. As examples, consider

$$\left|\varepsilon_{rel}\right|_k^{avg} = \frac{1}{L} \sum_{\ell=1}^{L} \left|\varepsilon_{rel}\left(\varphi_{k\ell}^U, \varphi_k^E\right)\right| , \tag{4-6}$$

$$\left|\varepsilon_{rel}\right|_\ell^{\max} = \frac{1}{N} \max_{k=1..N} \left|\varepsilon_{rel}\left(\varphi_{k\ell}^U, \varphi_k^E\right)\right| , \tag{4-7}$$

and

$$\left|\varepsilon_{rel}\right|^{avg} = \frac{1}{LN} \sum_{k=1}^{N} \sum_{\ell=1}^{L} \left|\varepsilon_{rel}\left(\varphi_{k\ell}^U, \varphi_k^E\right)\right| . \tag{4-8}$$

The first norm, $\left|\varepsilon_{rel}\right|_k^{avg}$, estimates the average (to some extent, typical) error in unfolding the flux in bin *k*. The second norm, $\left|\varepsilon_{rel}\right|_\ell^{\max}$, measures the worst-case error (among the bins) observed in the ℓ^{th} simulation. The third norm, $\left|\varepsilon_{rel}\right|^{avg}$, estimates the overall average error in unfolding with a particular set of response functions, $R_i^E(E)$, and with a specified level of random counting and/or calibration errors.

The Experiment Plan

This research is intended to demonstrate the need for unfolding and the need for an accurate, full-range recalibration of the HEEF. The plan is to conduct a series of numerical experiments which characterize the performance of the fluxmeter and how that performance depends upon the shape of the actual spectrum and the shape of the response functions. These numerical experiments utilize both idealized sets of response functions (described below), and the response functions from the first calibration or as extrapolated from the second, partial recalibration (as described in chapter 3). The research will explore the implications of the suspected shape of these new response functions.

The performance of these different response functions used to calculate the various unfolded fluxes can be compared and contrasted. In addition, the four types of error present in this process can be incorporated into this comparison. From this series of experimental case studies, conclusions can be drawn about the importance of finishing the second calibration and about the necessity of using a formal unfolding technique to calculate the unfolded electron fluxes.

There are four sets of numerical experiments; one set to answer each of the following questions:

1. Can an idealized set of response functions (described below) be used to approximate the set of response functions from the first calibration report?
2. Can an idealized set of response functions be used to approximate the set of response functions extrapolated from the second calibration report?
3. Can the first set of calibrated response functions be used as an approximation for the second set of response functions?

4. How do the presence of errors inherent to the unfolding technique, errors resulting from the discretization of the flux, errors in calibration, and errors in measurement alter the values of the unfolded electron fluxes?

Details of these tests are presented in the next chapter, but first, we define the idealized response functions.

An *idealized* response function, $R_i^I(E)$ is just

$$R_i^I(E) = R_{i,i}^C \chi_i(E). \tag{4-9}$$

This models the channel response as having constant sensitivity (at the average calibrated value) within the intended range of the channel, and no sensitivity outside that range. (This would be an ideal instrument channel.) This is equivalent to using a response matrix in which $R_{ik}^I = R_{ik}^C \delta_{ik}$. Then, the unfold reduces to solving an uncoupled set of equations:

$$y_i^M = \sum_k R_{ik}^I \varphi_k^{TU} = \sum_k R_{ik}^C \delta_{ik} \varphi_k^{TU} = R_{ii}^C \varphi_i^{TU}, \tag{4-10}$$

and hence

$$\varphi_i^{TU} = \frac{y_i^M}{R_{ii}^C}, \tag{4-11}$$

where φ_i^{TU} are the expansion coefficients for the *trivial unfold* spectrum:

$$\varphi^{TU}(E) = \sum_i \varphi_i^{TU} \chi_i(E). \tag{4-12}$$

V. Numerical Experimentation: Results and Observations

The numerical experiments contained within this chapter are divided into four different groups, where each group of experiments answers a specific goal listed in the statement of the problem found in chapter one. The first group explores the accuracy of unfolded fluxes when an idealized set of response functions is used as an approximation to the instrument sensitivities as detailed in the first calibration of the instrument. The second group is similar to the first, only the set of idealized sensitivities is used to approximate the set of response functions extrapolated from the partial recalibration of the instrument. The third set of numerical tests explores the implications of approximating the extrapolated response functions with the response functions derived from the first calibration of the HEEF. The last group quantifies the impact of the different sources of error inherent to measuring the electron flux. This includes error in the unfolding technique, error in the counts made by the HEEF, and error in the calibration of the instrument. In addition to the four groups of experiments, this chapter expands upon certain elements of the research as they pertain to the sets of various response functions. In particular, the motivation for using each set of response functions is discussed.

Each test group for the numerical experiments will contain the following information:

1. the sets of response functions used for folding and unfolding,
2. the rationale for using these sensitivities,
3. the known flux,
4. the calibration and counting error added to the process, if any,
5. a plot showing the continuous flux, the discretized flux, and the unfolded flux,
6. the error norms, and

7. a discussion of any observations and inferences.

This explanation for each specific experiment should provide the reader with a clear understanding of both the experimental technique and the results.

The Three Sets of Response Functions

It is possible to perform experiments with a wide variety of response functions, but only three sets are used in this research. These sets correspond to unfolding with an idealized sensitivity, unfolding with the sensitivities expressed in the first calibration of the fluxmeter, and unfolding with the sensitivities extrapolated from the partial recalibration of the HEEF. As one experiments with the numerical unfolding procedure it becomes very clear that these three sets of response functions cannot be used as approximations for each other. Unfolding with the correct instrument sensitivity is critical for calculating accurate electron fluxes.

The first set of sensitivities is the set of idealized response functions. These are the response functions one would use for doing a trivial unfold. The response matrix, R_{ik}^{I}, has only diagonal elements; all other entries are zero. Each diagonal element (for purposes of this research) has the same value as the corresponding diagonal element in the response matrix for the first calibration and the energy values for the wide bin boundaries are also the same as for the first calibration. The mathematical relations for calculating this matrix are found in chapter three. In essence, the idealized sensitivity could be used to approximate the set of response functions obtained from the first calibration (which has a diagonally dominant response matrix), but whether or not this approximation is valid remains to be seen. These idealized response functions are shown

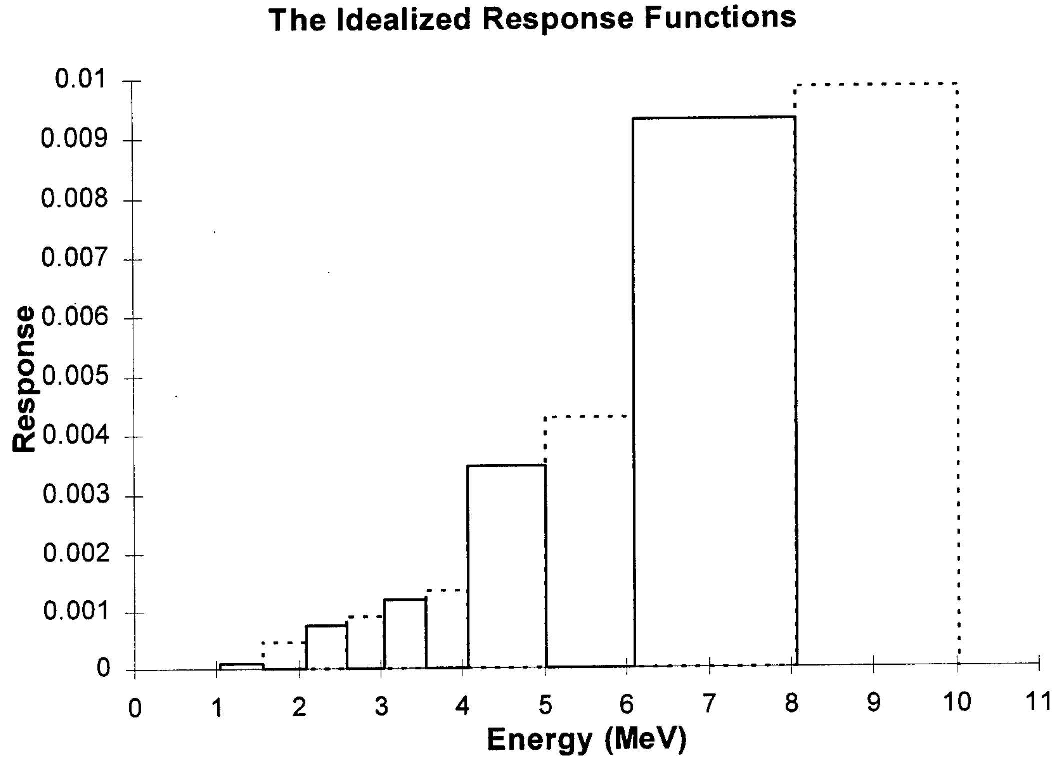

Figure 8. The set of idealized response functions.

in figure 8. Idealized response functions based on the second calibration are used to provide its trivial unfold.

The second set of response functions used in the experimentation are the sensitivities quantified in the first calibration of the fluxmeter. Chapter three explains the calculation of this response, and it includes analytical expressions for the geometric factor, the relative response, and the absolute response. The third set of response functions are the sensitivities extrapolated from the partial recalibration. Once again, chapter three details the development and lists the set of mathematical expressions used to calculate the sensitivities. Although the figures showing the specific responses for the two calibrations are not included here, there is a different way to view this. Each

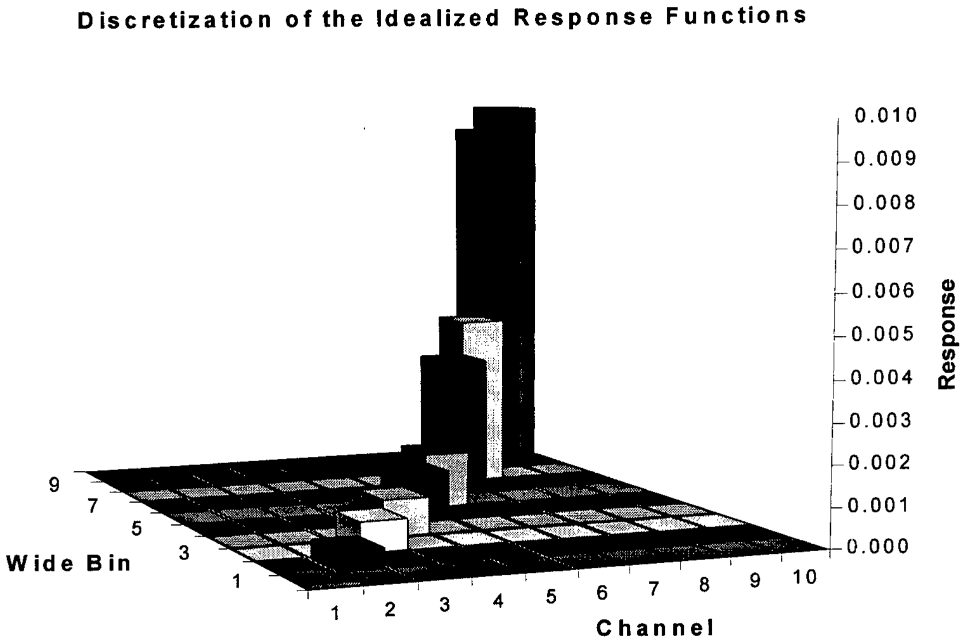

Figure 9. The discretization of the idealized response functions.

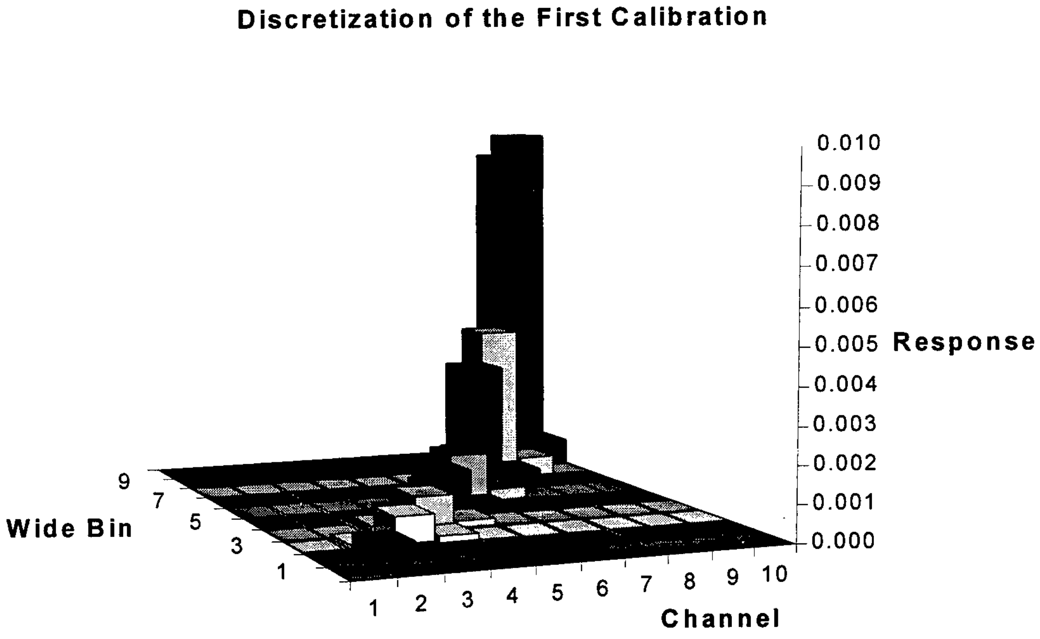

Figure 10. The discretization of the response functions for the first calibration.

response matrix for each set of response functions represents a discretization of the sensitivity, and this can be shown in three dimensional plots of R_{ik}. Recall that i is the instrument channel and k is the wide bin. Figures 9, 10, and 11 show the progression from a diagonal matrix (idealized responses) to a diagonally dominant matrix (first calibration) to a nearly upper-triangular matrix (the partial recalibration).

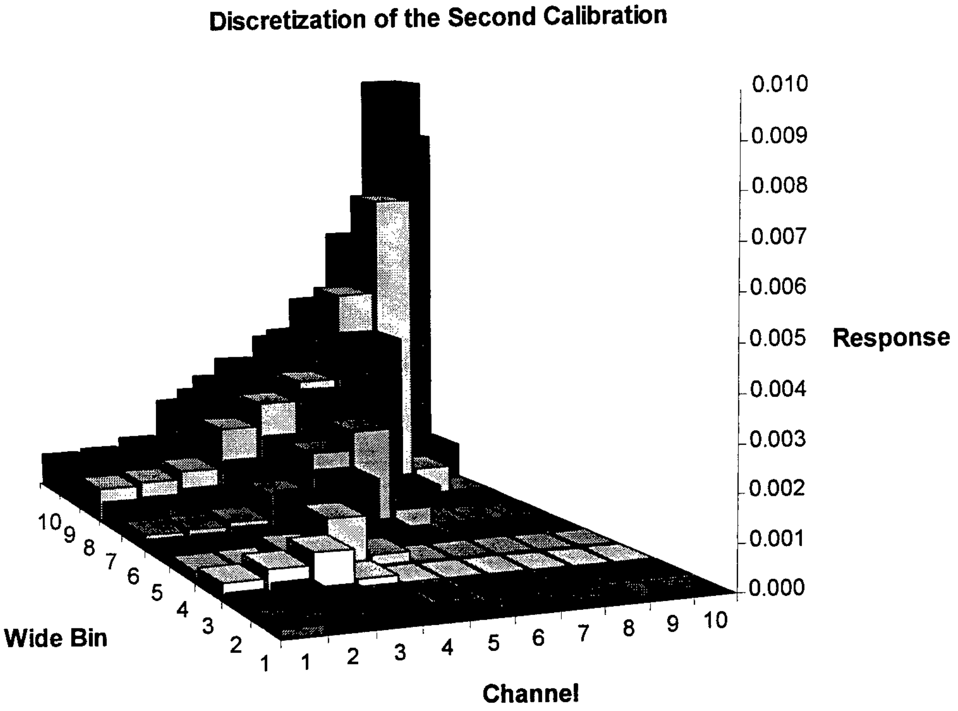

Figure 11. The discretization of the response functions for the second calibration.

The three dimensional plots of the response matrices capture the essence of this unfolding problem, for the most important result of the research is that the shape of the response function plays a dominant role in the unfolding technique. Accurate determination of the shape of the response functions is essential for successful unfolding.

This is why the partial recalibration of the HEEF must be completed, and the numerical experimentation will justify this conclusion.

Approximating the First Calibration with a Set of Idealized Response Functions

This experimental group has only two test cases. It shows that using the idealized set of response functions as an approximation for the response functions derived from the first calibration of the instrument introduces an unacceptably large amount of error into the unfolded flux (as compared with the discretized flux). The unfolded electron fluxes shown here simulate the results one would obtain by attempting a trivial unfold (by using a direct inversion of the idealized response matrix).

Test Case 1:

The set of response functions used for folding is the set from the first calibration. The set of response functions used for unfolding is the set of idealized sensitivities. This simulates using a trivial unfold to approximate unfolding with responses obtained from the first calibration of the instrument. The exact flux is

$$\varphi^{E}(E) = E^{-p} \tag{5-1}$$

where p is 4 and E is the energy of the electron. There is no counting or calibration error incorporated into this unfold. Figure 12 shows the continuous flux, the discretized flux, and the unfolded flux.

The histogram plot shows large differences between the discretized flux and the unfolded flux. Note how the continuous flux does not pass through several of the unfolded flux bin values. The extent of the error is difficult to measure on the logarithmic scale, but the maximum error in the ten wide bins is 213 % (the ratio of the difference between the unfolded and discretized flux to the discretized flux) and the average error is approximately 107 %. These error ranges are unacceptably large, and so using the idealized response functions as an approximation for the response functions derived from the first calibration (for this type of known flux) is ill-advised.

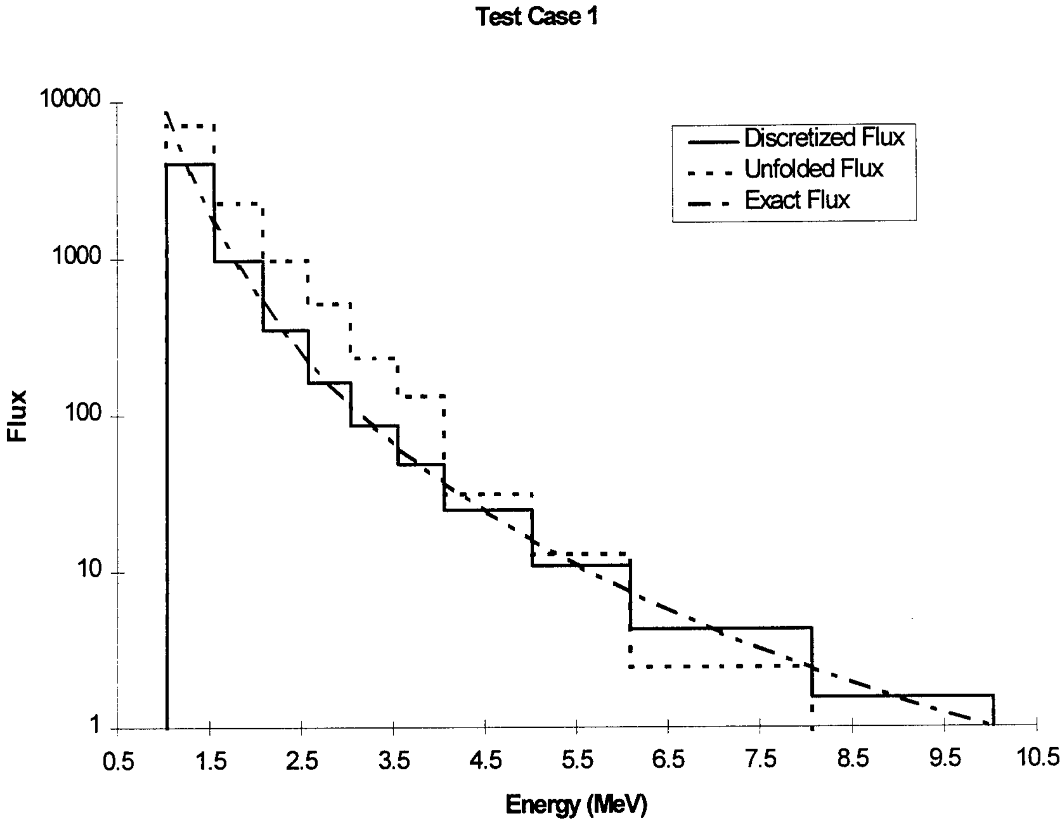

Figure 12. The flux values for test case 1.

Test Case 2:

The set of response functions used for folding is the set from the first calibration. The set of response functions used for unfolding is the set of idealized sensitivities. This simulates using a trivial unfold to approximate unfolding with responses obtained from the first calibration of the instrument. This time, however, the exact flux is

$$\varphi^E(E) = \left(\frac{1.128}{\tau}\right)\cdot\left(\frac{E}{\tau}\right)^{1/2}\cdot\exp\left(\frac{-E}{\tau}\right) \tag{5-2}$$

where τ is the temperature (in MeV) and E is the energy of the electron. The peak in this Maxwell-Boltzmann distribution is at 0.5 MeV; the temperature is $\tau = 1$ MeV. There is no counting or calibration error incorporated into this unfold. Figure 13 shows the continuous flux, the discretized flux, and the unfolded flux.

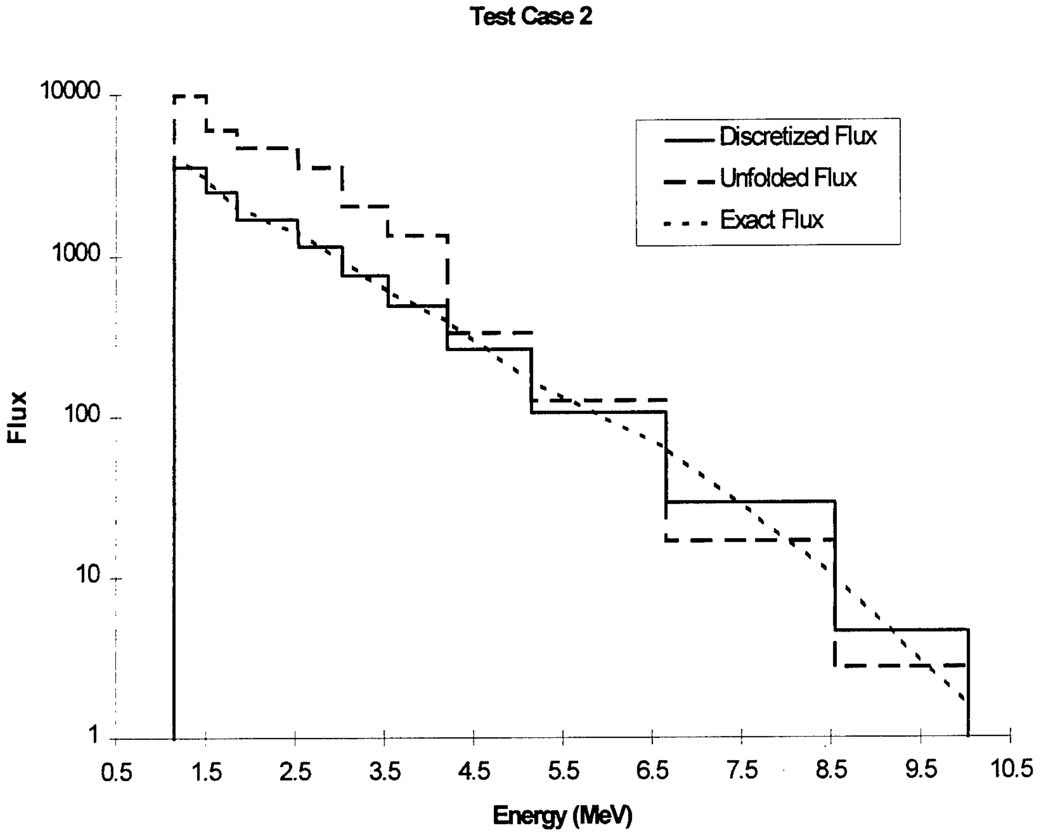

Figure 13. The flux values for test case 2.

Once again, the histogram plot shows large differences between the discretized flux and the unfolded flux. The maximum error in the ten wide bins is 206% and the average error is approximately 116 %. The results are not strongly dependent on the functional form of the flux, so using the idealized response functions as an approximation for the response functions derived from the first calibration is not recommended.

Approximating the Second Calibration with a Set of Idealized Response Functions

This experimental group only requires two test cases. It shows that using the idealized set of response functions as an approximation for the response functions derived from the partial recalibration of the instrument introduces an unacceptably large amount of error into the unfolded flux (as compared with the discretized flux). The unfolded electron fluxes shown here simulate the results one would obtain by attempting a trivial unfold (by using a direct inversion of the diagonalized response matrix) with a response matrix that is nearly upper triangular.

Test Case 3:

The set of response functions used for folding is the set from the partial recalibration. The set of response functions used for unfolding is the set of idealized sensitivities. This simulates using a trivial unfold to approximate unfolding with responses obtained from the second calibration of the instrument. The exact flux is

$$\varphi^{E}(E) = E^{-p} \tag{5-3}$$

where p is 4 and E is the energy of the electron. There is no counting or calibration error incorporated into this unfold. Figure 14 shows the continuous flux, the discretized flux, and the unfolded flux.

If the shape of the response functions hinted at in the recalibration is correct, an idealized set of response functions (a diagonal response matrix) cannot be used for unfolding the flux. The maximum error in the ten wide bins is 777 % and the average error is approximately 291 %. Obviously, the off-diagonal elements of the response matrix cannot be ignored. This test case shows the importance of completing the partial recalibration of the HEEF. The shape of the response functions must be accurately determined.

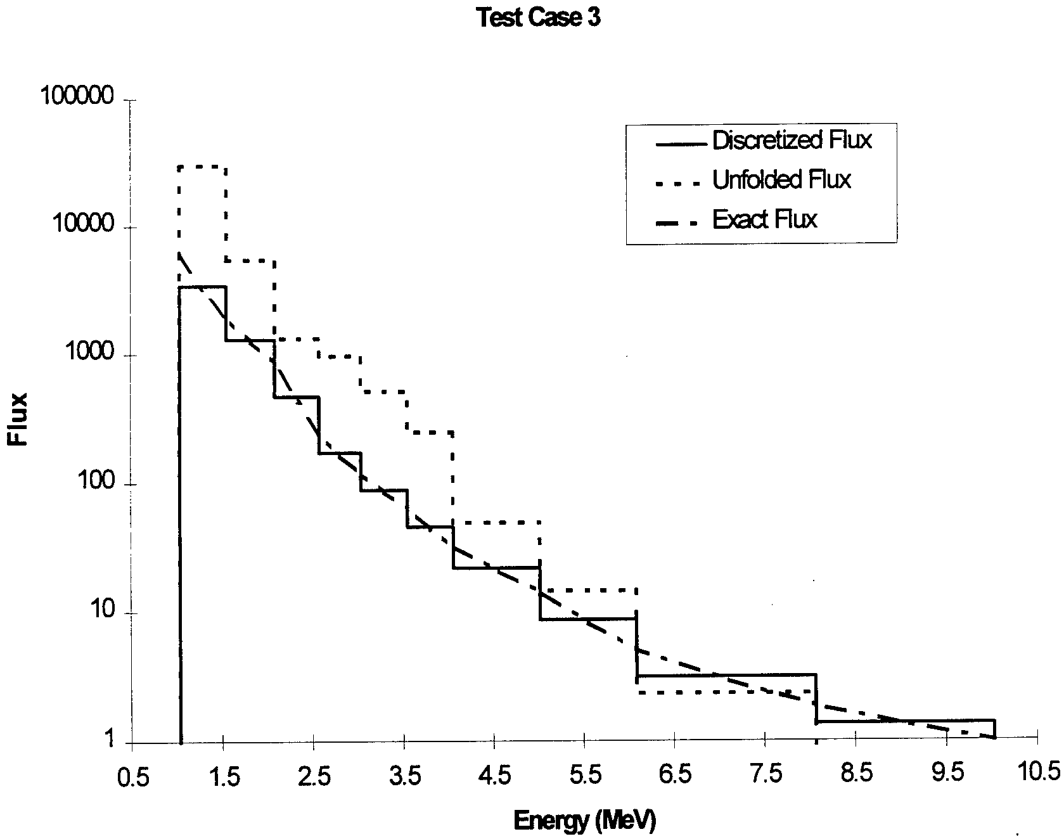

Figure 14. The flux values for test case 3.

Test Case 4:

The set of response functions used for folding is the set from the second calibration. The set of response functions used for unfolding is the set of idealized sensitivities. This simulates using a trivial unfold to approximate unfolding with responses obtained from the partial recalibration of the instrument. This time, however, the exact flux is

$$\varphi^E(E) = \left(\frac{1.128}{\tau}\right) \cdot \left(\frac{E}{\tau}\right)^{1/2} \cdot \exp\left(\frac{-E}{\tau}\right) \qquad (5\text{-}4)$$

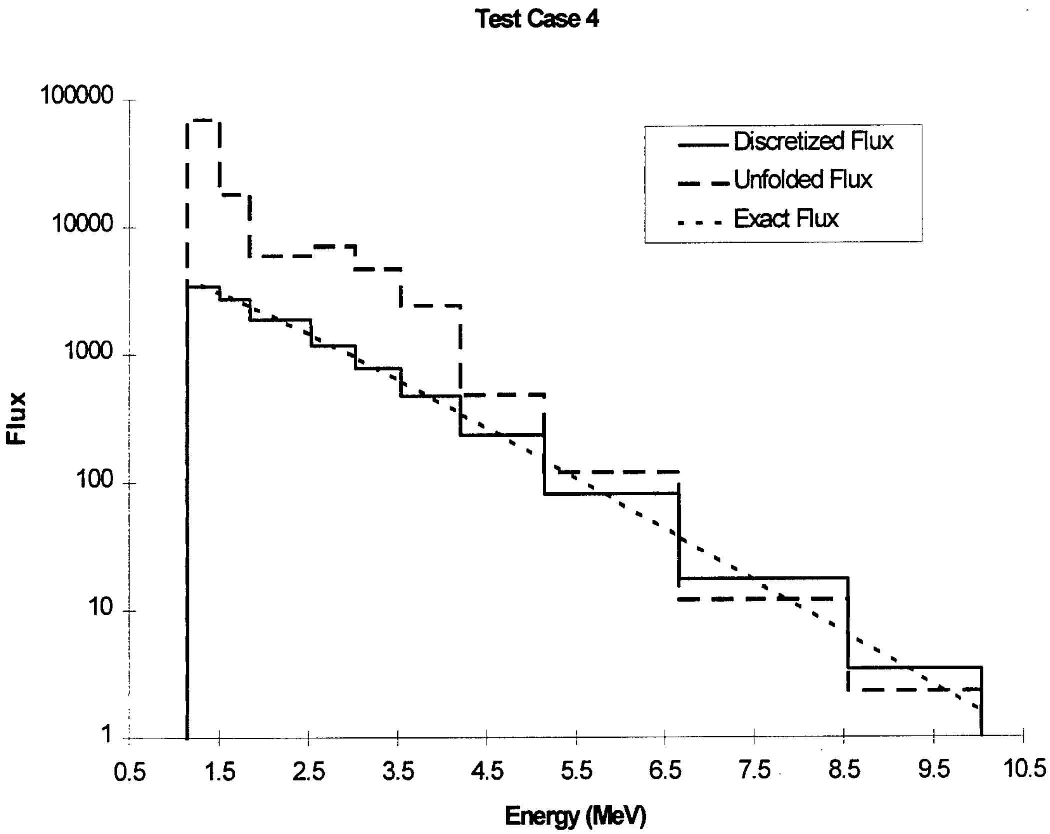

Figure 15. The flux values for test case 4.

where τ is the temperature (in MeV) and E is the energy of the electron. The peak in this Maxwell-Boltzmann distribution is at 0.5 MeV; the temperature is $\tau = 1\ \mathrm{MeV}$. There is no counting or calibration error incorporated into this unfold. Figure 15 shows the continuous flux, the discretized flux, and the unfolded flux.

This test case is included to show that the results in test case 3 are not unique to the functional form of the flux. The maximum error in the ten wide bins is 1903 % and the average error is approximately 436 %. The idealized response functions simply cannot be used as an approximation for the instrument sensitivities extrapolated from the second calibration.

Approximating the Second Calibration with the First Calibration

This experimental group, like the first two, only requires two test cases. It shows that using the first calibrated set of response functions as an approximation for the response functions extrapolated from the partial recalibration of the instrument introduces large amounts of error into the unfolded flux (as compared with the discretized flux). These numerical experiments show that there are important differences in the shapes of the two sets of HEEF response functions. In essence, a diagonally dominant response matrix cannot be substituted for one that is nearly upper triangular. If the second calibration of the instrument is accurate, than the unfolding computations must be performed with response functions which accurately reflect the second set of instrument sensitivities.

Test Case 5:

The set of response functions used for folding is the set from the partial recalibration. The set of response functions used for unfolding is the set from the first calibration of the instrument. This simulates using the first set to approximate unfolding with responses obtained from the second set. The exact flux is

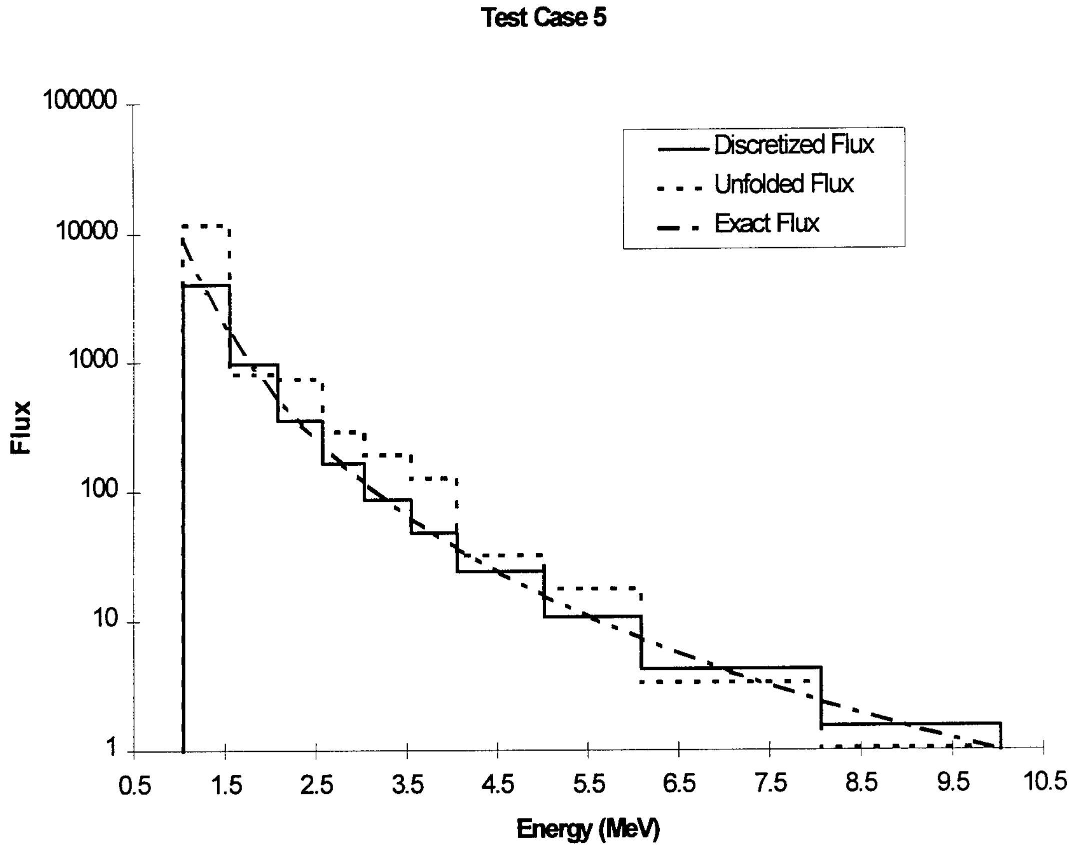

Figure 16. The flux values for test case 5.

$$\varphi^{E}(E) = E^{-p} \tag{5-5}$$

where p is 4 and E is the energy of the electron. There is no counting or calibration error incorporated into this unfold. Figure 16 shows the continuous flux, the discretized flux, and the unfolded flux.

The motivation for test cases 5 and 6 is proving the need for a complete recalibration of the fluxmeter. As is readily apparent, the two response matrices cannot be interchanged. The maximum error in the ten wide bins is 190 % and the average error is approximately 82 %. The shape of the response functions is critical, and a large high energy tail in the sensitivity cannot be ignored, even though the spectrum decreases rapidly with increasing energy.

Test Case 6:

The set of response functions used for folding is the set from the second calibration. The set of response functions used for unfolding is the set from the first calibration of the HEEF. This simulates using the first set to approximate unfolding with responses obtained from the second set. This time, however, the exact flux is

$$\varphi^E(E)=\left(\frac{1.128}{\tau}\right)\cdot\left(\frac{E}{\tau}\right)^{1/2}\cdot\exp\left(\frac{-E}{\tau}\right) \tag{5-6}$$

where τ is the temperature (in MeV) and E is the energy of the electron. The peak in this Maxwell-Boltzmann distribution is at 0.5 MeV; the temperature is $\tau = 1\ \text{MeV}$. There is no counting or calibration error incorporated into this unfold. Figure 17 shows the continuous flux, the discretized flux, and the unfolded flux.

Test case 6 demonstrates that the results in test case 5 are not a function of the flux. The maximum error in the ten wide bins is 562 % and the average error is approximately 125 %.

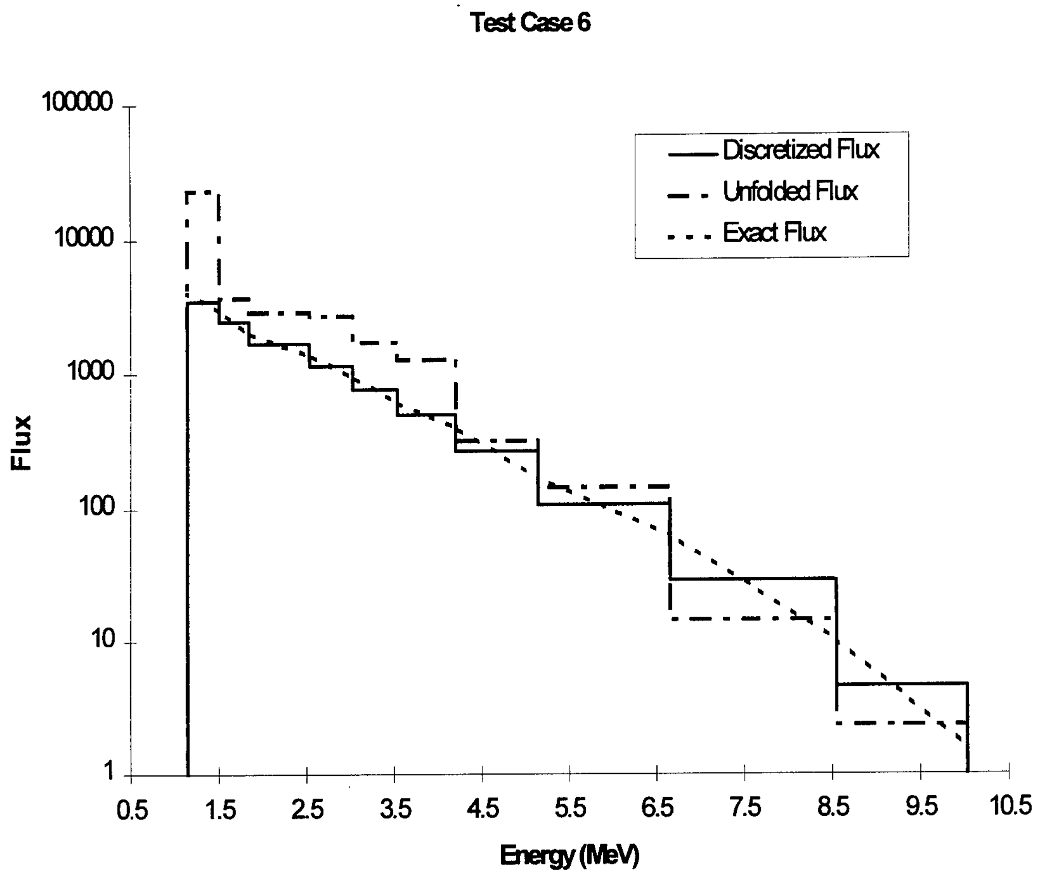

Figure 17. The flux values for test case 6.

Incorporating the Various Types of Error into the Unfolding Calculations

This experimental group contains eight test cases. These eight scenarios show the impact of discretization error, error inherent to the unfolding technique, calibration error, and counting error. From these experiments one can explore how coupled and de-coupled instrument sensitivities perform under the influence of different types of error. When will this error propagate and grow? In addition, these numerical experiments show which type of error dominates the unfolding calculations.

Test Case 7:

The set of response functions used for both folding and unfolding is the set from the first calibration. This experiment explores the error introduced through the discretization of the flux and through the ill-conditioned nature of the unfolding problem. The exact flux is

$$\varphi^{E}(E) = E^{-p} \tag{5-7}$$

where p is 4 and E is the energy of the electron. There is no counting or calibration error incorporated into this unfold. Figure 18 shows the continuous flux, the discretized flux, and the unfolded flux.

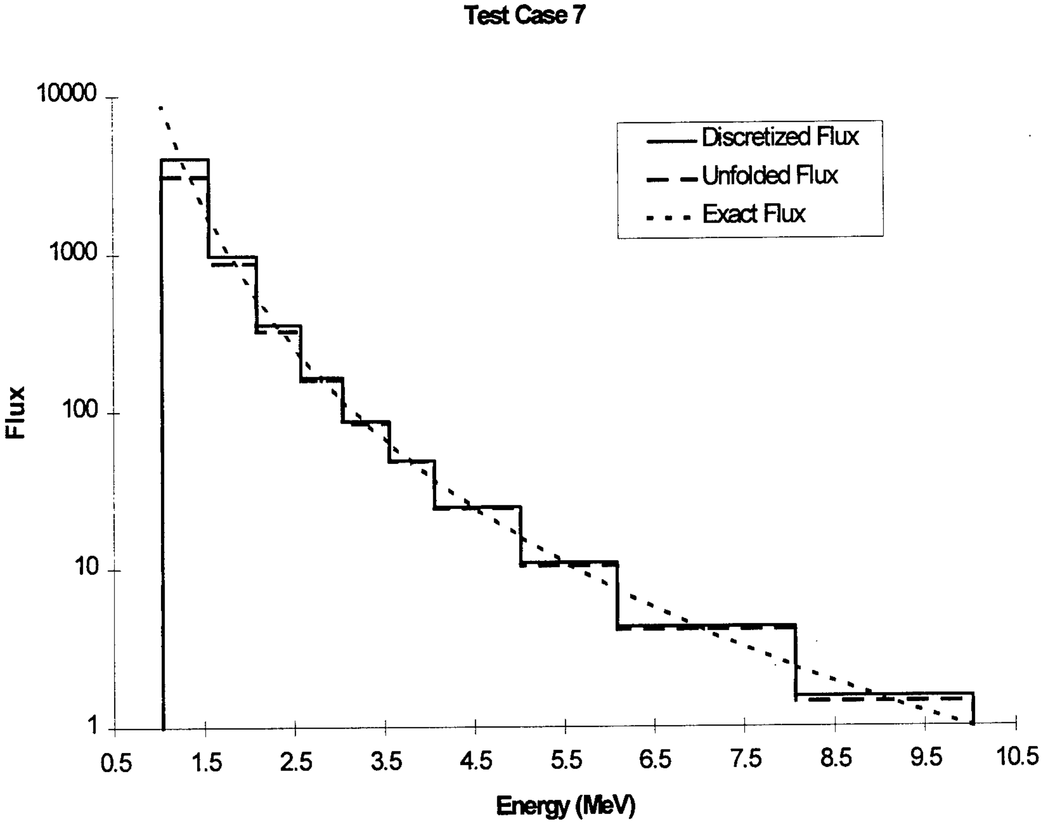

Figure 18. The flux values for test case 7.

Test cases 7 and 8 are designed to show how coupling in the responses between successive instrument channels can alter the accuracy of the unfolded fluxes. In test case 7, which uses the response functions from the first calibration, there is very little coupling between successive channels. The maximum error in the ten unfolded flux bins is 23.1 %. This error occurs in the first channel. Refer to figure 5 in chapter three which shows the response functions for the first calibration (as calculated by the code) and note the extent of the overlap of instrument responses between channels one and two. The second largest error is 10.4 %, and this is found in the second channel. The average error in the wide bins is 6.8 % (4.3 % without the first two channels). This experiment shows that coupling can introduce error into the unfold; the error in channels one and two compared to the errors in the other eight channels (not as strongly coupled as the first two) demonstrate this nicely.

Test case 8:

The set of response functions used for both folding and unfolding is the set from the partial recalibration. This experiment explores the error introduced through the discretization of the flux and through the ill-conditioned nature of the unfolding problem. The exact flux is

$$\varphi^{E}(E) = E^{-p} \tag{5-8}$$

where p is 4 and E is the energy of the electron. There is no counting or calibration error incorporated into this unfold. Figure 19 shows the continuous flux, the discretized flux, and the unfolded flux.

As in the experiment for test case 7, the concern in this example is coupling in the response functions between successive instrument channels. If one compares the response functions (as calculated by the code) for the first calibration and the partial recalibration (shown in figure 7 in chapter three), it appears the responses in the second calibration are more strongly coupled. The maximum error in this test case is 13.9 %,

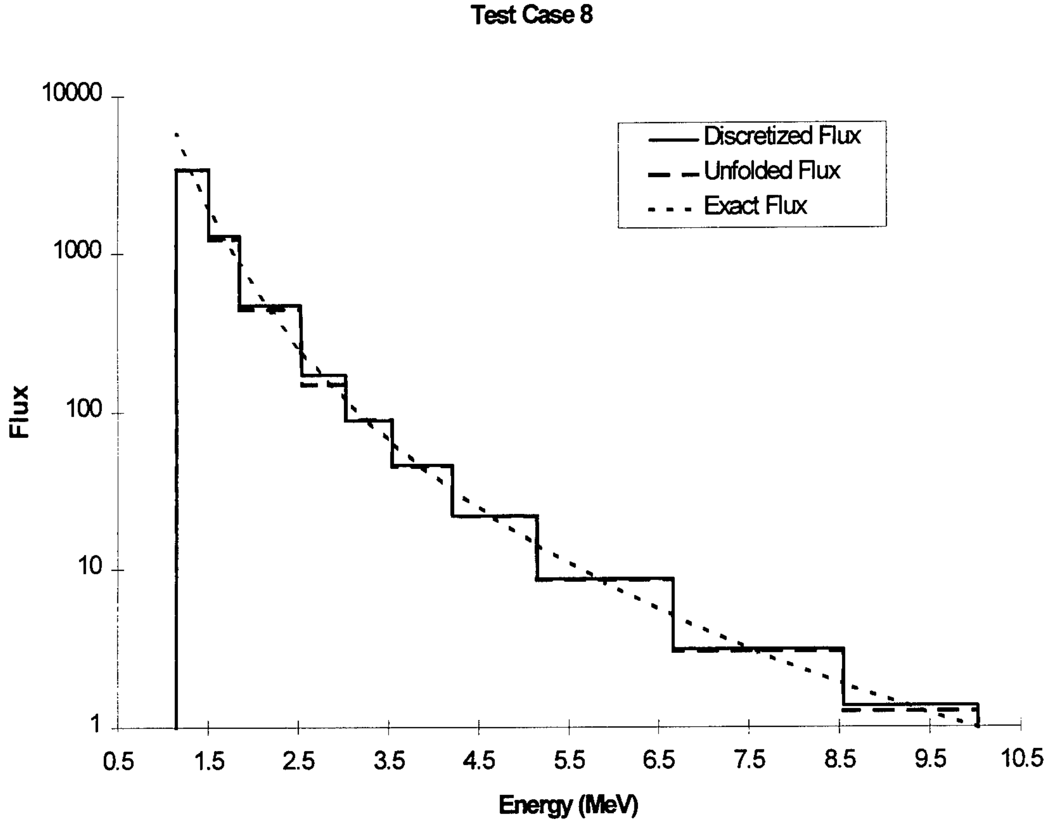

Figure 19. The flux values for test case 8.

and the average error is 4.4 %. At first, this result seems contradictory. Both the maximum error and the average error in test case 8 are less than test case 7. Note, however, that the coupling in the first two channels in the first calibration is very strong. This is where the largest errors occur. If these two channels are removed from the average error calculation, the average error in the unfolded flux produced by the second set of response functions is slightly larger. Another point worth mentioning is the values assigned to the tails in the second set of response functions. This information was extrapolated from the figure in reference 5 which showed the absolute responses derived from the partial recalibration. If the value of the absolute responses for these tails are different from what the researchers extrapolated (specifically, if that value is greater), the degree of coupling will be significantly enhanced. This is one of the reasons why finishing the second calibration of the HEEF is so important.

Test Case 9:

The set of response functions used for both folding and unfolding is the set from the first calibration. Calibration errors of 5% (by which we mean $\varepsilon^{CAL} = 5\%$) and 1% counting errors (by which we mean $\varepsilon^{count} = 1\%$) are simulated in this unfolding calculation. This experiment explores the impact counting and calibration error have on fluxes unfolded by response functions which exhibit a small degree of coupling between successive instrument channels. The exact flux is

$$\varphi^{E}(E) = E^{-p} \tag{5-9}$$

where p is 4 and E is the energy of the electron. Figure 20 shows the continuous flux, the discretized flux, and the unfolded flux.

For this type of unfold, the maximum error in the wide bins is approximately 20 % and the average error is approximately 6 %. This is not significantly different than the error results stated in test case seven. If one wide bin has a large error in the unfolded flux (resulting from calibration or counting error), this error cannot grow and propagate through other channels because the instrument channels are de-coupled. As an end result,

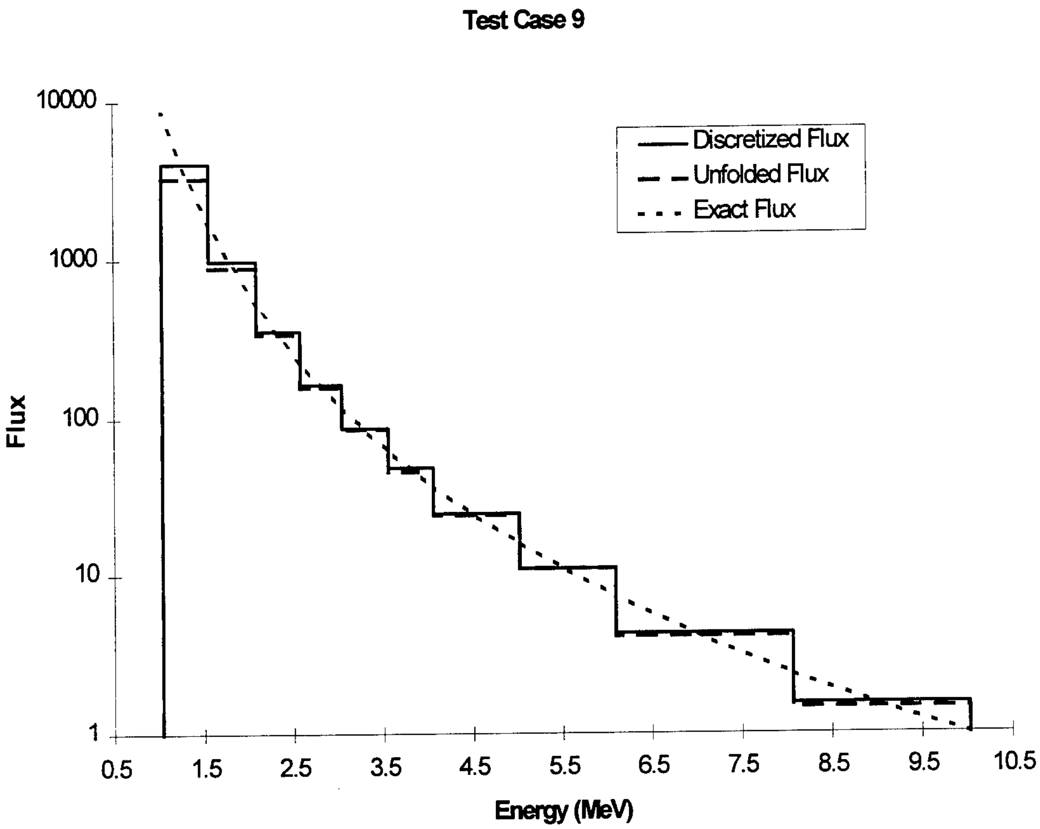

Figure 20. The flux values for test case 9.

the errors in test cases seven and nine are approximately the same. Note that a 5 % error in the calibration and an 1 % counting error in the signal counts represents a best case scenario: the calibration was correctly performed and the instrument was operating properly. Remember that the signal counts have already been taken, and an 1 % error in the channel with the maximum counts as a best case was an input from the sponsor of the research [4].

Test Case 10:

The set of response functions used for both folding and unfolding is the set from the partial recalibration. There is a calibration error of 5 % and a counting error of 1 % incorporated into this unfolding calculation. This experiment explores the impact counting

and calibration error have on fluxes unfolded by response functions which exhibit a larger degree of coupling between successive instrument channels. The exact flux is

$$\varphi^E(E) = E^{-p} \tag{5-10}$$

where p is 4 and E is the energy of the electron. Figure 21 shows the continuous flux, the discretized flux, and the unfolded flux.

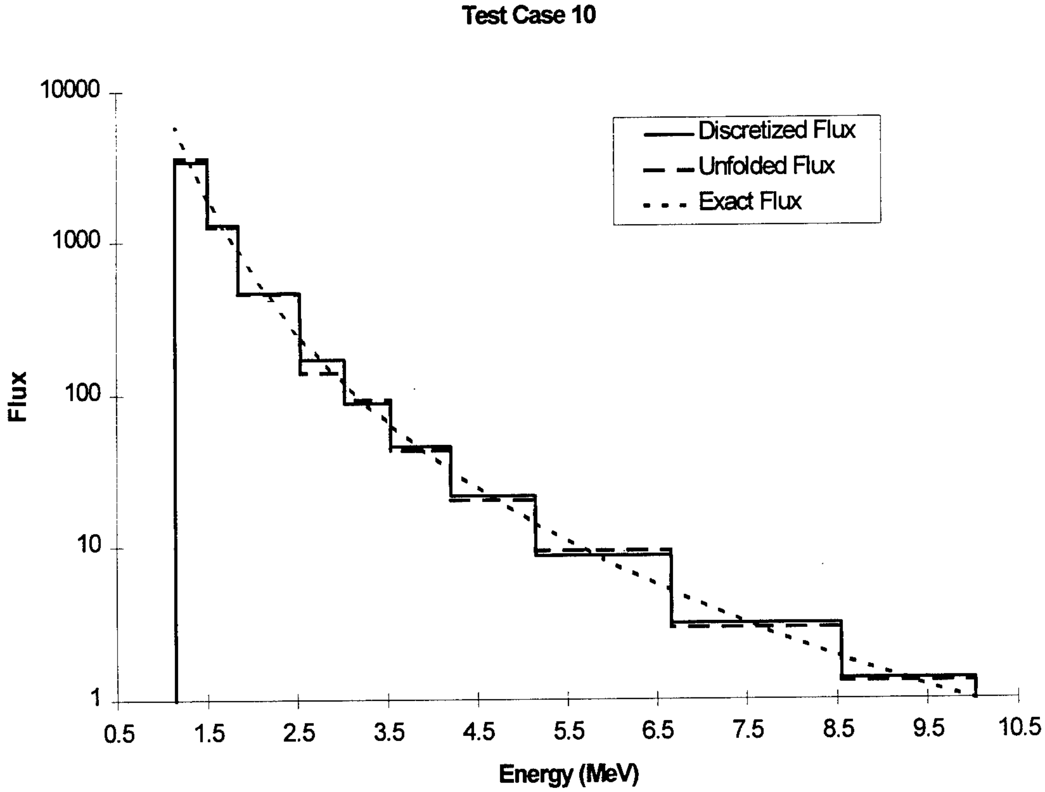

Figure 21. The flux values for test case 10.

The value for the average wide bin unfolded flux error is approximately 7 %, with a maximum error in the wide bins of approximately 20 %. Although subtle, this test case shows an interesting result. When no calibration error or counting error was incorporated into the unfold, one could argue that the second set of response functions actually performed better than the set from the first calibration. Once calibration and counting error enter into the scenario, the performance changes. With error values representative of a best case calibration and instrument performance, the maximum errors are about the same but the average error resulting from use of the second set of response functions is larger. The presence of the high energy tails in the partial recalibration are beginning to be felt.

There are two points which deserve further development. First, the performance of these sets of response functions depends on the functional form of the electron flux. With the Fortran 90 code, it is possible to generate hundreds of test cases but this does not necessarily mean every possible test case should be included. To keep the analysis manageable, the test cases incorporating error analysis from the unfolding technique, the instrument calibration, and the measured signals only use one type of flux. The spectrum shown in equations (5-7) through (5-14) was selected based on input from the sponsor at Phillips Laboratory. It is the flux which, at the present time, best models the measured electron spectra in the radiation belts [5].

The second point concerns the generation of random noise for the different test cases. This random noise depends, in part, on seeds input by the person using the Fortran 90 code. One test case cannot constitute a conclusive data set, and so ten test cases are calculated for each test case which uses calibration and counting error. The data in the ten test cases are used to calculate the error ratios shown in figures 22, 23, 26, 27, 30, and 31. These figures show the average absolute value of error ratio and the maximum absolute value of error ratio by trial spectra and by energy bin for test cases 10, 12, and 14. Once again, to keep the data analysis manageable, the statistical plots are limited to test cases incorporating signal and calibration error in conjunction with the second set of response functions.

In these statistical studies of the propagation of error, we use error ratio rather than per cent relative error. We make this choice because error ratio treats high and low errors symmetrically. For example, an error by a factor of two high is a relative error of 100%, while an error by a factor of two low is a relative error of -50%. Yet both errors are the same size (factor of two) on a logarithmic plot, and both are represented by the

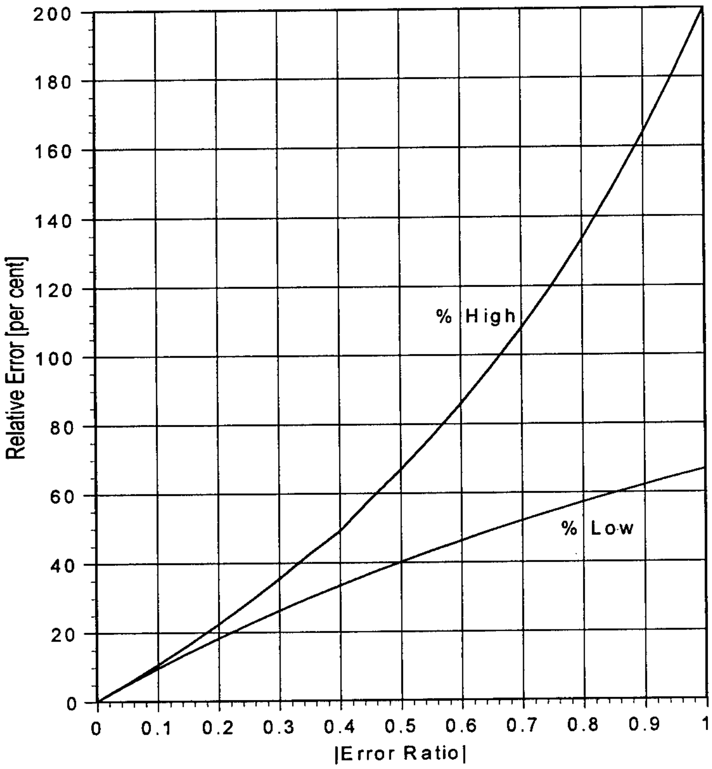

Figure 22. Percentage high and low relative errors vs. error ratio

same magnitude error ratio: a factor of two high/low is an error ratio of $\pm 2/3$. Figure 22 shows the relative errors in per cent (high and low) corresponding to error ratios in the range $0 \le |\varepsilon_{rel}| \le 1$. It may be helpful in interpreting the results reported below.

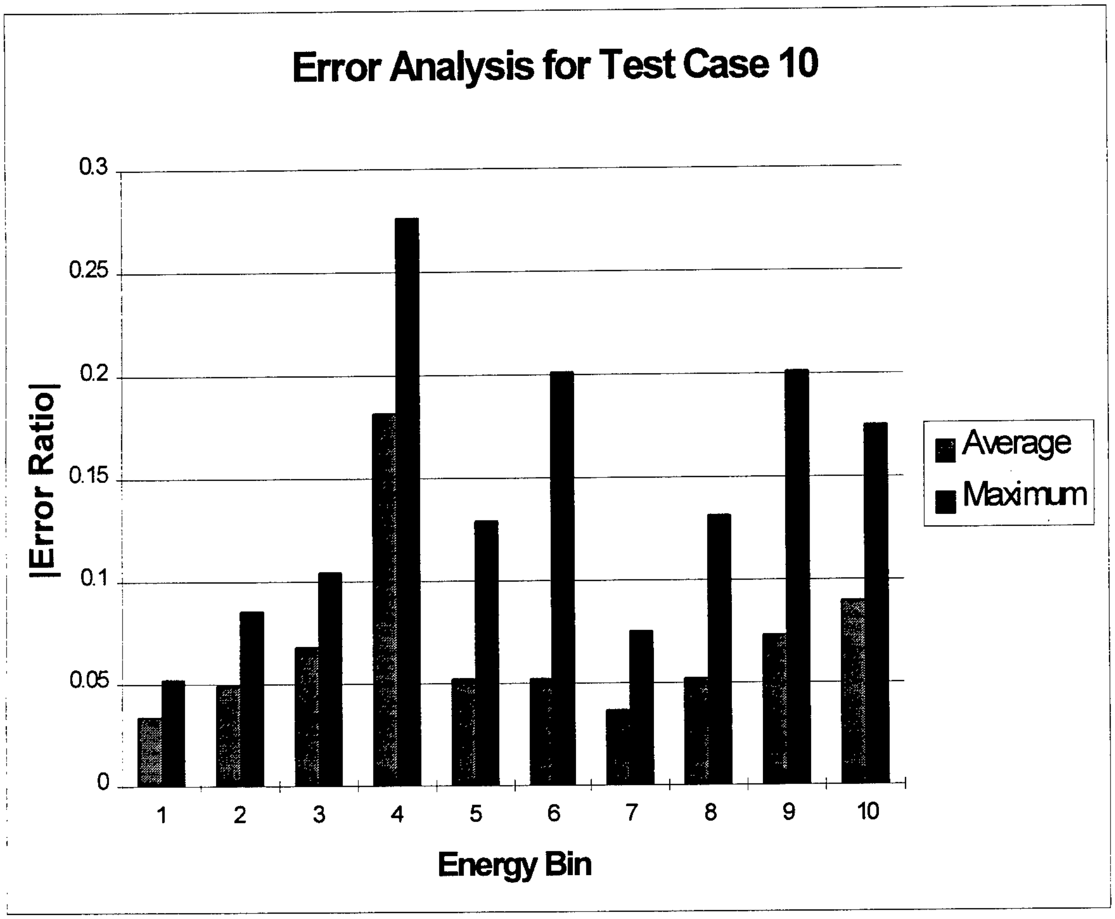

Figure 23. A statistical error analysis, by energy bin, for test case 10.

Figure 23 shows an average relative error in the unfolded flux, by energy bin, ranging from 0.03 to 0.17 (remember, these relative error numbers do not correspond to error percentages). Maximum relative errors in the unfolded flux, by energy bin, range from 0.05 to 0.28. Figure 24 shows the relative errors not by energy bin, but by generated test case (trial spectrum). In this example, the average relative error varies from 0.05 to 0.10. The maximum relative errors have a low value of 0.10 and a high value of 0.28. In essence, these figures show that when no calibration or signal error are incorporated into the unfold, the second set of response functions performs slightly better than the first set. When experiments which simulate a best case scenario for calibration and signal error are computed, the second set of response functions now performs slightly worse than the first set. This is a direct result of the degree of coupling between successive instrument channels. The response functions from the partial recalibration exhibit stronger coupling,

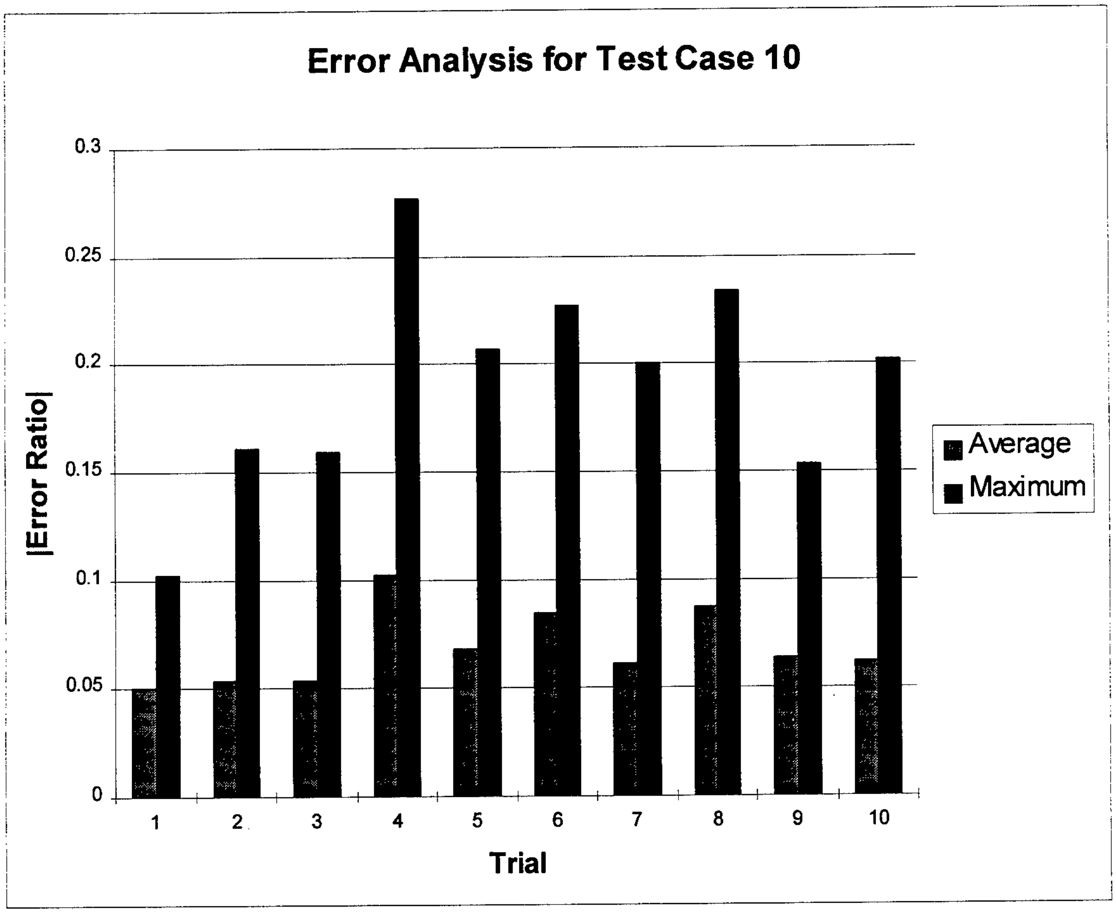

Figure 24. A statistical error analysis, by trial spectra, for test case 10.

and so signal and calibration error can propagate and grow from the high energy channels to the low energy channels (because the response function tails go from low to high energy). Compare the magnitudes of the relative error in test cases 10, 12, and 14 as the amount of counting and calibration error increase.

Test Case 11:

The set of response functions used for both folding and unfolding is the set from the first calibration. There is a calibration error of 20 % and a counting error of 1 %

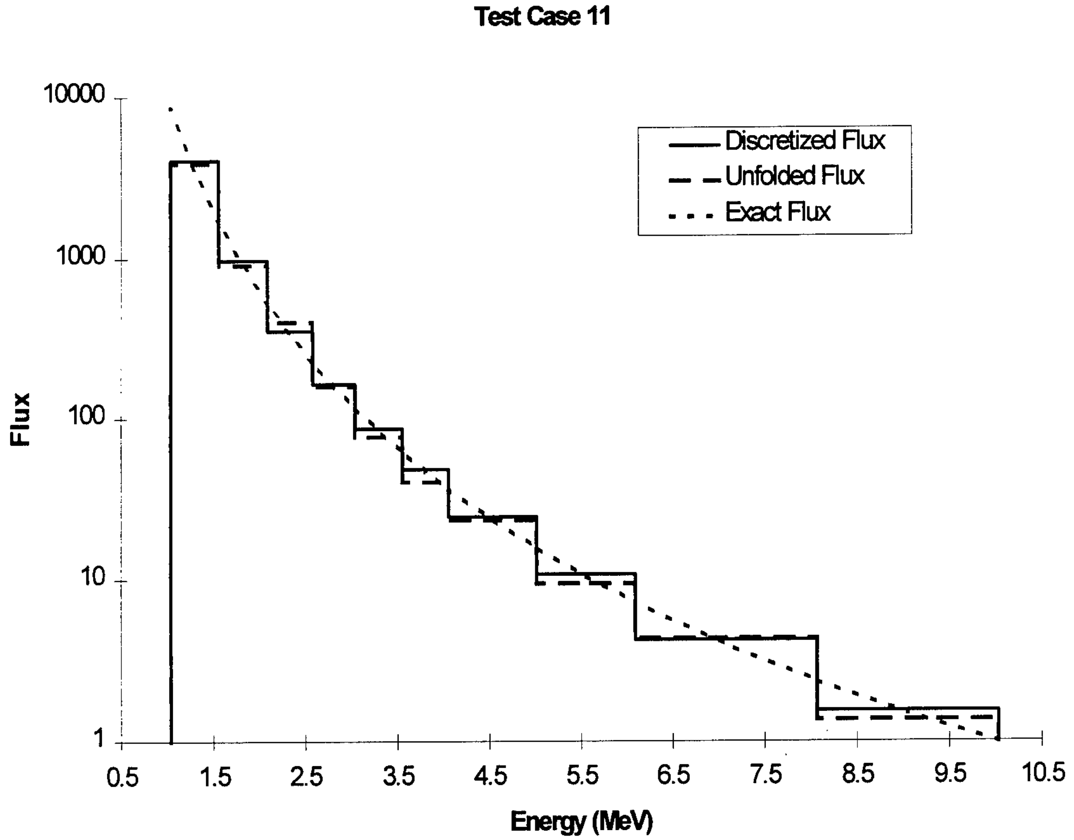

Figure 25. The flux values for test case 11.

incorporated into this unfolding calculation. This experiment explores the impact counting and calibration error have on fluxes unfolded by response functions which exhibit a small degree of coupling between successive instrument channels. The exact flux is

$$\varphi^{E}(E) = E^{-p} \tag{5-11}$$

where p is 4 and E is the energy of the electron. Figure 25 shows the continuous flux, the discretized flux, and the unfolded flux.

Test case 11 simulates an unfold with a best case counting error and a calibration error indicative of the accuracy obtained with the first calibration of the instrument. The maximum error in the wide bins ranges from 15 % to 20 %. This is very similar to test cases 7 and 9. The average error in the wide bins is approximately 9 %. Even when simulating a worst case scenario in the calibration, the decoupling of the response functions derived from the first calibration does not allow the error to grow and propagate. The maximum errors are the same as when no calibration or counting error were added to the unfolding process. The average errors have increased slightly from test case 7 to test case 9 to test case 11, but they are still less than ten percent.

Test Case 12:

The set of response functions used for both folding and unfolding is the set from the partial recalibration. There is a calibration error of 20 % and a counting error of 1 % incorporated into this unfolding calculation. This experiment explores the impact counting and calibration error have on fluxes unfolded by response functions which exhibit a larger degree of coupling between successive instrument channels. The exact flux is

$$\varphi^{E}(E)=E^{-p} \qquad (5\text{-}12)$$

where p is 4 and E is the energy of the electron. Figure 26 shows the continuous flux, the discretized flux, and the unfolded flux.

The maximum error per wide bin in test case 12 is approximately 25 % and the average error per wide bin is 11 %. Under the same conditions as those in test case 11,

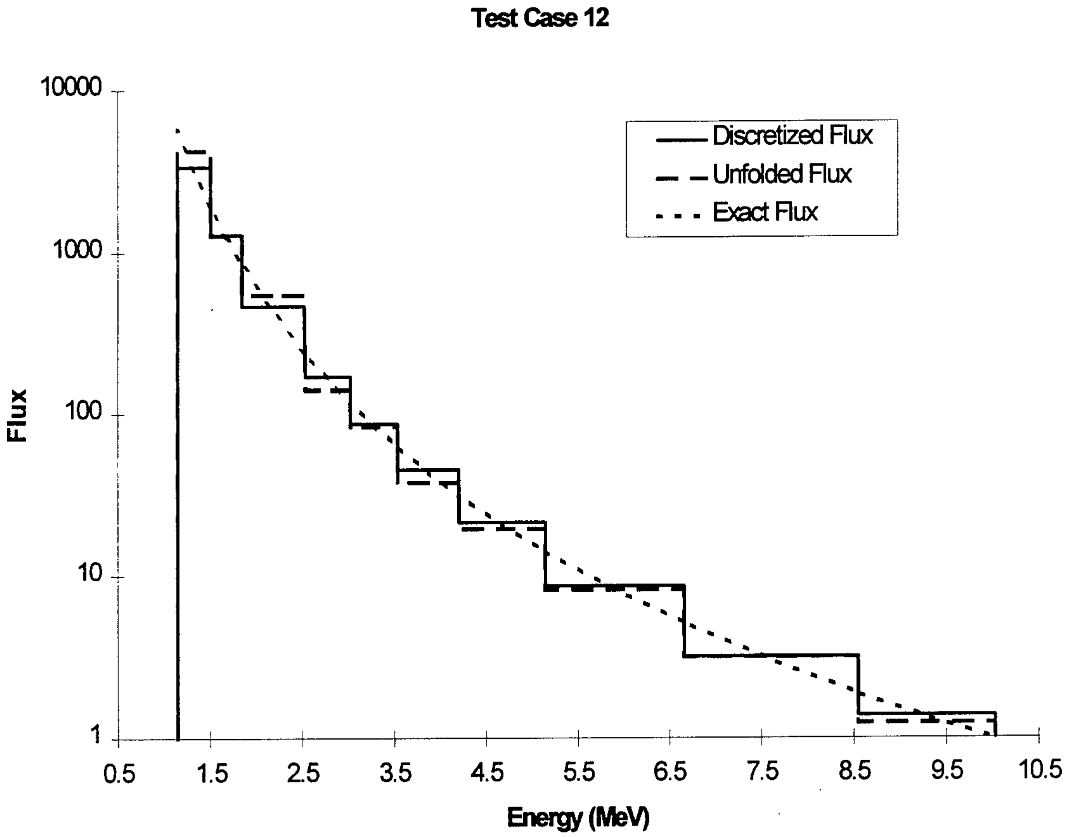

Figure 26. The flux values for test case 12.

the second set of response functions is performing noticeably worse than the first set. The shape of the response functions is critical to the unfolding process, and the presence of coupling in the responses between the different instrument channels is the most important attribute of the functional shape.

The relative error plot, by wide energy bin, in figure 27 shows the average relative error lies within the range of 0.07 to 0.20. The maximum relative error ranges from 0.15 to 0.30. Figure 28 shows the relative average error, by trial spectrum, varies from 0.08 to 0.17. The maximum relative error has a minimum value of 0.20 and a maximum value of

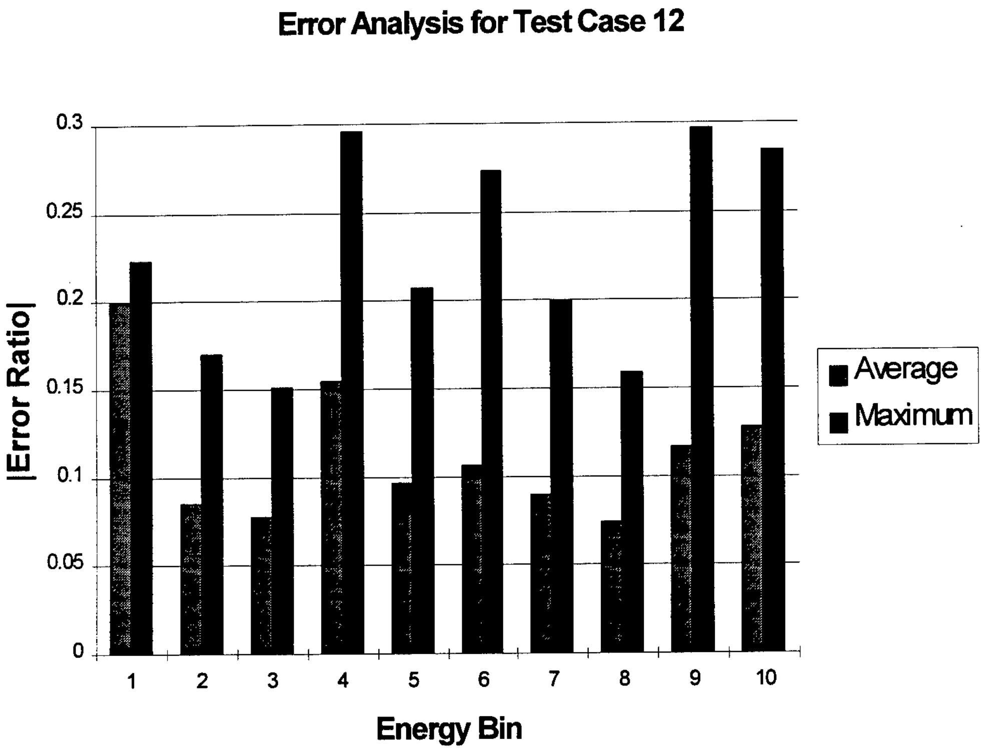

Figure 27. A statistical error analysis, by energy bin, for test case 12.

0.29. This information shows that increasing the calibration error does increase the error in the unfolded flux, but the increase is not excessively disproportionate. Case study 12 shows calibration error does not dominate the process of unfolding the electron flux.

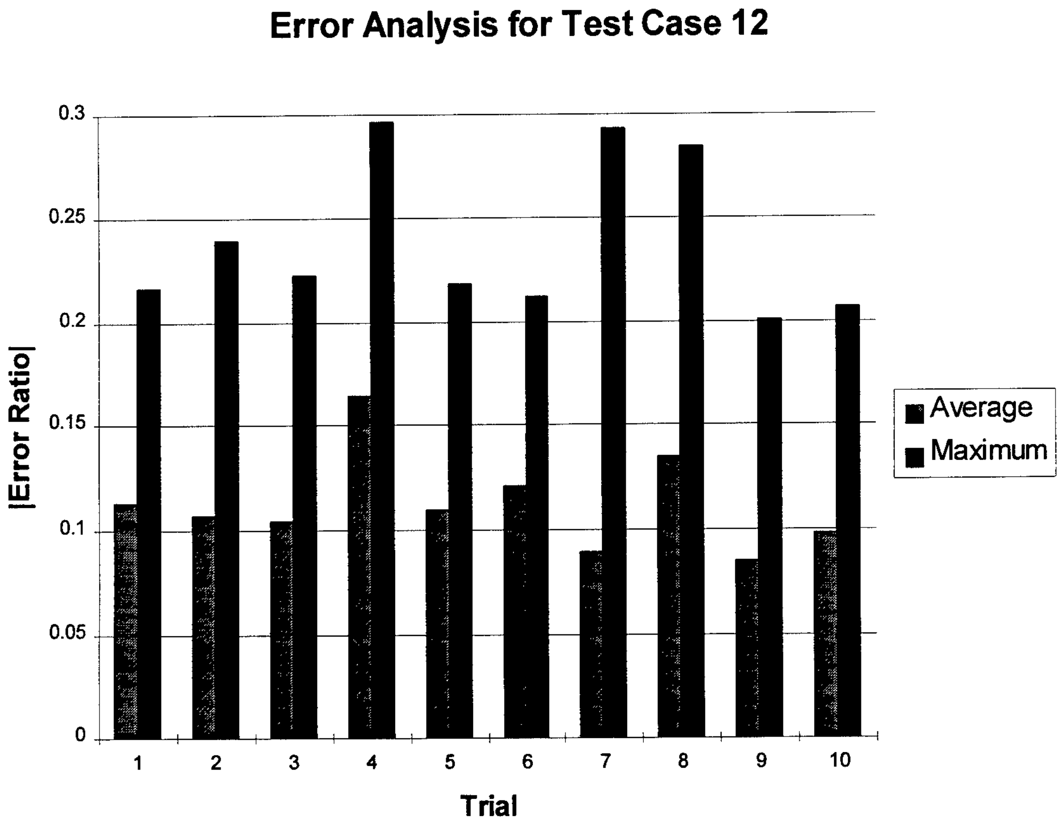

Figure 28. A statistical error analysis, by trial spectra, for test case 12.

Test Case 13:

The set of response functions used for both folding and unfolding is the set from the first calibration. There is a calibration error of 20 % and a counting error of 5 % incorporated into this unfolding calculation. This experiment explores the impact counting and calibration error have on fluxes unfolded by response functions which exhibit a small degree of coupling between successive instrument channels. The exact flux is

$$\varphi^E(E) = E^{-p} \tag{5-13}$$

where p is 4 and E is the energy of the electron. Figure 29 shows the continuous flux, the discretized flux, and the unfolded flux.

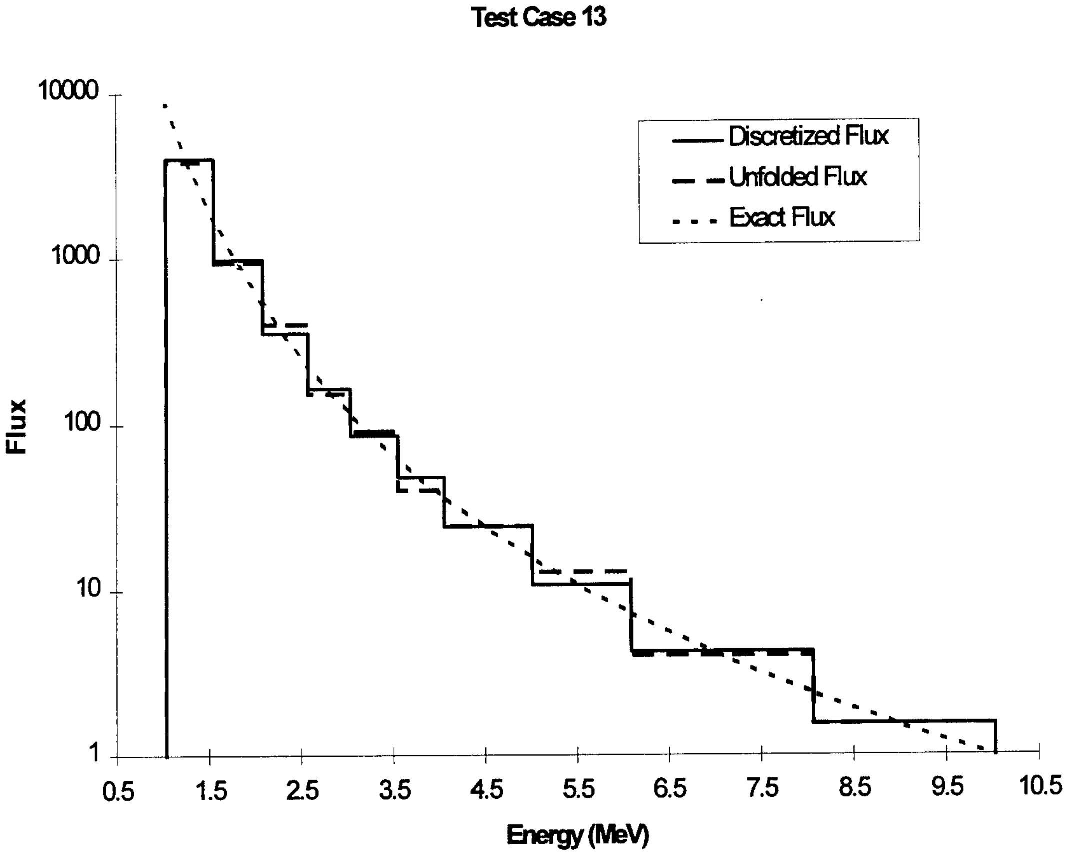

Figure 29. The flux values for test case 13.

This test case represents a worst case scenario for unfolding with HEEF measurements. A calibration error of 20 % simulates calibrating with an accuracy on order of that obtained in the first calibration of the instrument. An error of 5 % in the channel with the most counts is indicative of poor (but not unreasonably poor) performance of the fluxmeter. The maximum error in the unfolded flux bins varies from 20 % to 25 %. The average error is approximately 10 %. Considering the large amount of error incorporated into this unfold, the results seem quite good, for average errors are still relatively small. Even the maximum errors are not that excessive. Response functions which have a small degree of coupling between successive energy channels perform well in calculating unfolded fluxes. Unfortunately, the partial recalibration shows this de-coupling to be unrealistically optimistic, for the responses have tails. In addition, test cases 5 and 6 showed that the first calibrated responses cannot be used to approximate the response functions extrapolated from the partial recalibration.

Test Case 14:

The set of response functions used for both folding and unfolding is the set from the partial recalibration. There is a calibration error of 20 % and a counting error of 5 % incorporated into this unfolding calculation. This experiment explores the impact counting

and calibration error have on fluxes unfolded by response functions which exhibit a larger degree of coupling between successive instrument channels. The exact flux is

$$\varphi^E(E) = E^{-p} \tag{5-14}$$

where p is 4 and E is the energy of the electron. Figure 30 shows the continuous flux, the discretized flux, and the unfolded flux.

The maximum error per wide bin in test case 12 is approximately 40 % to 45 % and the average error per wide bin is 23 %. Under the same conditions as those in test case

13, the second set of response functions is performing significantly worse than the first set. These two test cases provide even more evidence that the shape of the response functions will impact the unfolding process. The more coupled the response functions, the more ill-conditioned the unfold and the greater the error introduced into the unfolded high energy electron flux.

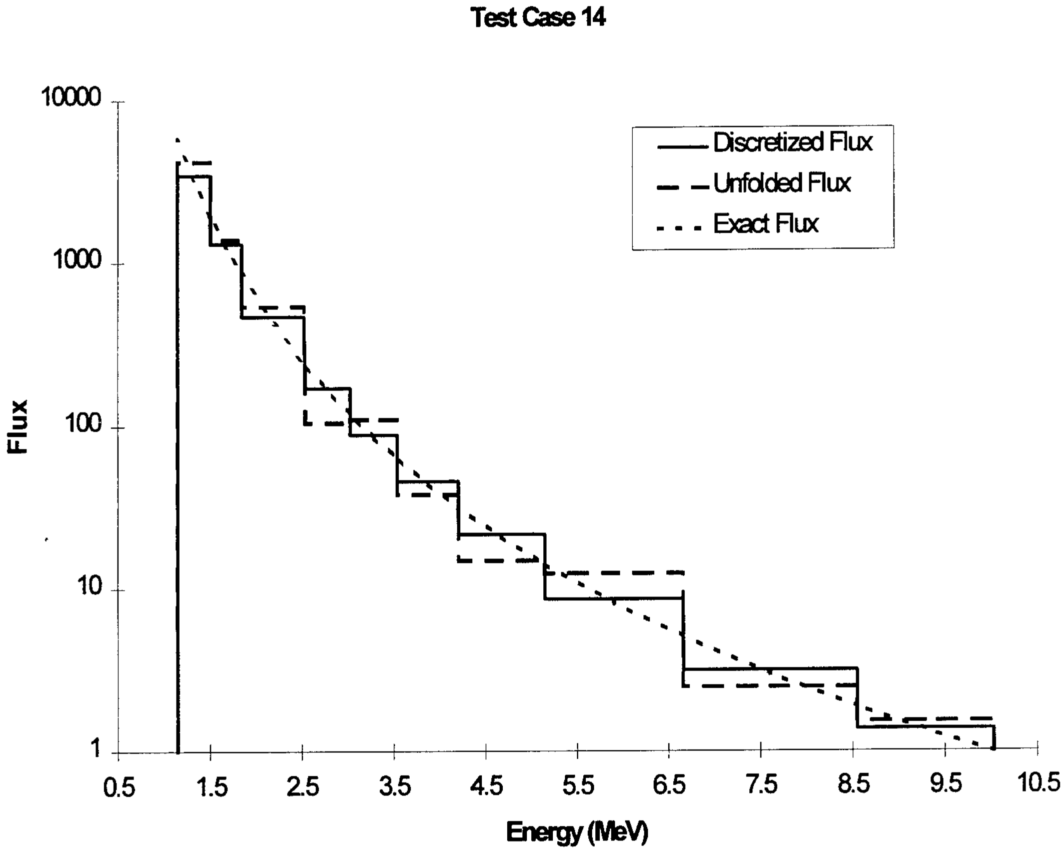

Figure 30. The flux values for test case 14.

The relative error plot, by wide energy bin, in figure 31 shows the average relative error lies within the range of 0.14 to 0.44. The maximum relative error ranges from 0.22

to 1.22. Figure 28 shows the relative average error, by trial spectrum, varies from 0.10 to 0.44. The maximum relative error has a minimum value of 0.18 and a maximum value of 1.22. This information shows that error in the signal counts is the dominant error in the unfolding process. Coupled response functions open the door to error growth and propagation. Even if the unfolding process itself (without the addition of counting or calibration error) does not introduce a large degree of error into the unfolded flux, the coupling is still there. A calibration error of 20 % and a counting error of 5 % essentially destroy the unfold if the response functions are coupled.

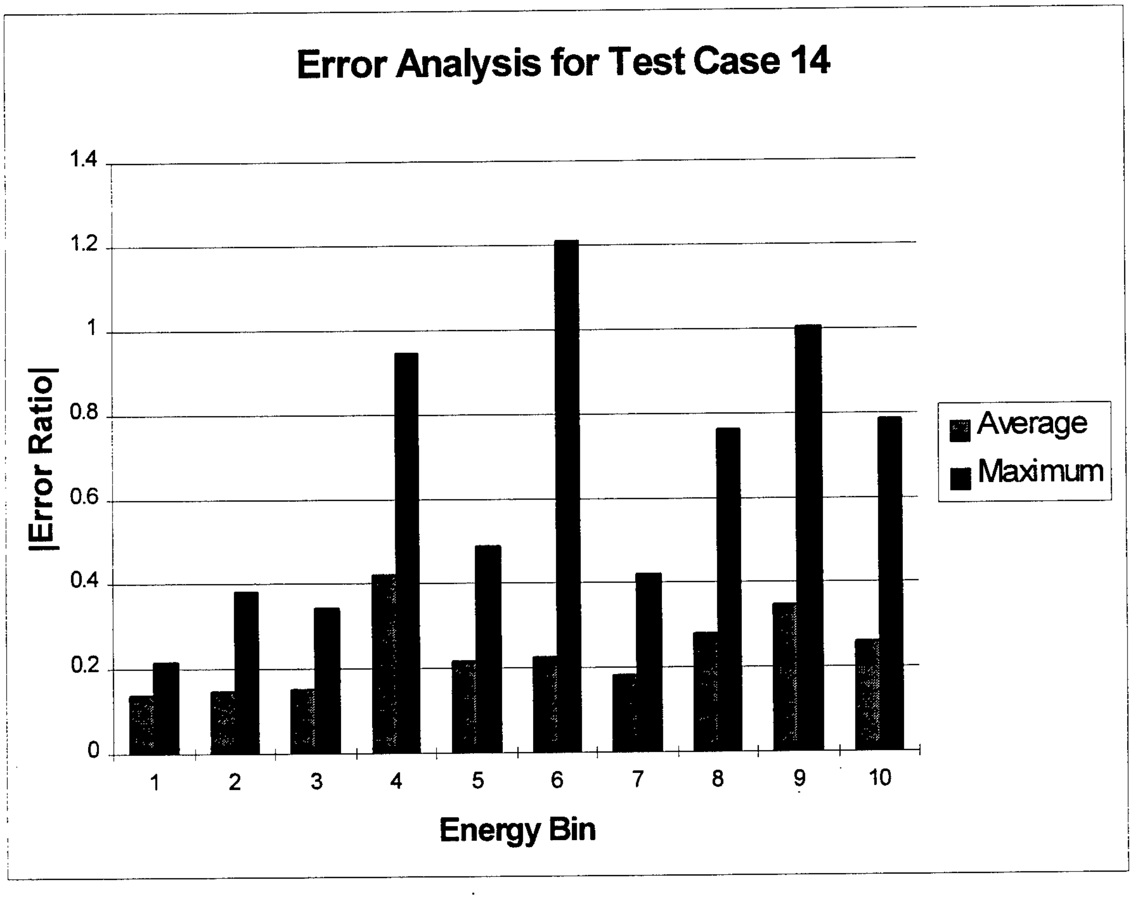

Figure 31. A statistical error analysis, by energy bin, for test case 14.

The test cases in this chapter were selected to demonstrate several different points. First, the shape of the response function is important. Idealized approximations cannot be used for response functions which are strongly coupled. The unfolding process cannot be done in a trivial manner. Second, coupled response functions allow growth and propagation of error. Increasing the calibration and counting error had little impact on fluxes unfolded with the first set of response functions. This was not the case with the second set, for by increasing the error in the calibration and in the signal counts the confidence level in the values of the unfolded flux collapsed. Third, it appears that the error in the signal counts has a stronger impact on the unfolded fluxes than does that of the calibration error. The reader must be careful in drawing this last conclusion,

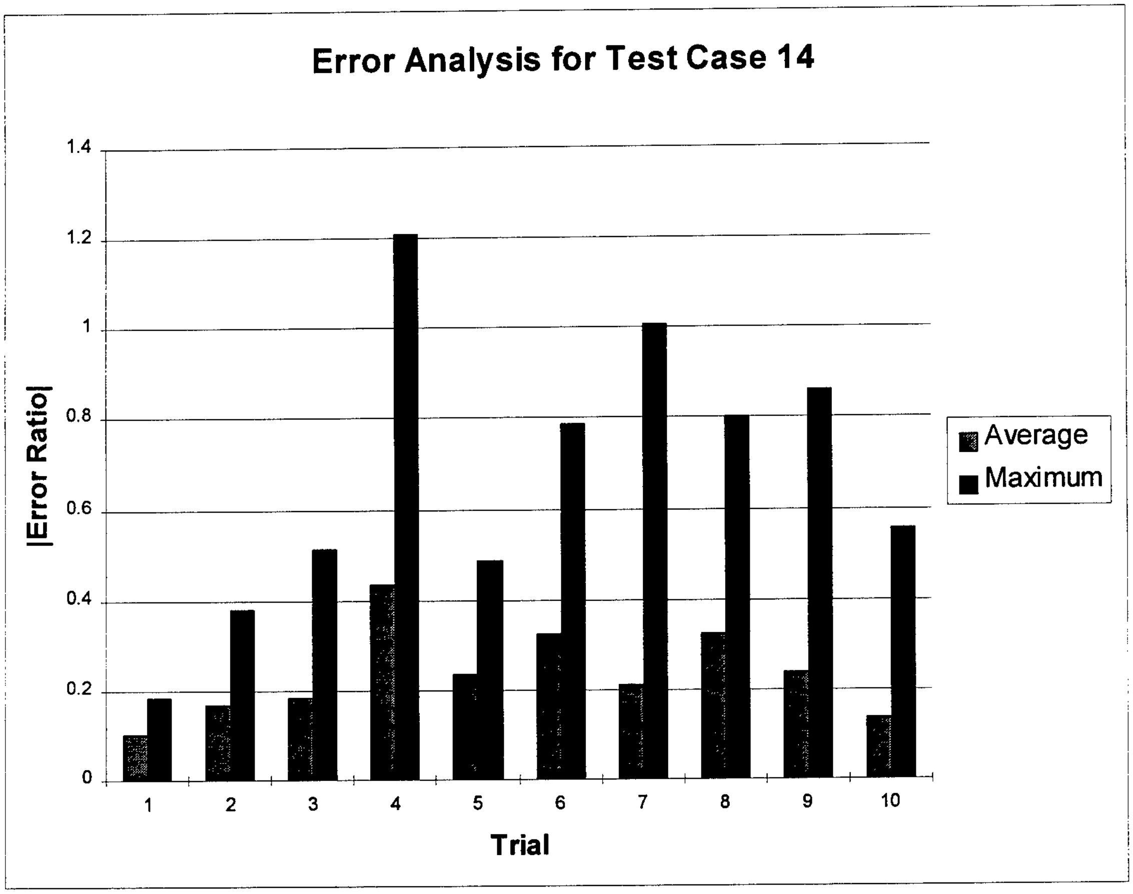

Figure 32. A statistical error analysis, by trial spectra, for test case 14.

for the impact of the signal error is dependent upon the form of the flux. If an electron spectrum decays rapidly with higher energies, the errors in the counts made in the higher energy bins will be relatively larger, as shown in table 3. Since the response functions are coupled from higher energies to lower energies, a large error in the counts in a high energy bin may propagate and grow to the lower energy bins. The important point is that the correct response functions must be used, they cannot be approximated with idealized cases. In addition, the calibration error and the counting error must be minimized, or the error growth and propagation will destroy the unfold.

		Relative Standard Deviation [%]	
Channel	Normalized Signal	$\varepsilon_{count} = 1\%$	$\varepsilon_{count} = 5\%$
1	1.000	1.00	5.00
2	0.775	1.14	5.68
3	0.618	1.27	6.36
4	0.380	1.62	8.11
5	0.281	1.89	9.43
6	0.196	2.26	11.30
7	0.144	2.63	13.16
8	0.083	3.47	17.34
9	0.037	5.22	26.09
10	0.011	9.41	47.06

Table 3. Expected Counting Errors for $\varphi^E(E) \sim E^{-4}$ with Second Calibration

The next chapter will summarize the results shown in the test cases and will present general observations and conclusions. Also, recommendations for further action and research will be discussed.

VI Conclusions and Recommendations

The research conducted for this thesis can be summarized by stating the key results in a table which includes observations from all 14 test cases. In addition to the table, observations can be organized into inferences about trends discovered during the research. In particular, comments on the shape of the response functions are provided. From this summary final conclusions will be drawn. The chapter ends with a section detailing recommendations for further action.

Summary of Results

The results obtained from the numerical experiments in chapter five are summarized in Table 4. The number of the test case in the table corresponds to the number of the test case in chapter five. The flux used for the experiment is denoted by

$$\varphi(E) = E^{-4} \equiv \varphi_{pl} \tag{6-1}$$

or

$$\varphi(E) = \left(\frac{1.128}{\tau}\right) \cdot \left(\frac{E}{\tau}\right)^{1/2} \cdot \exp\left(\frac{-E}{\tau}\right) \equiv \varphi_{mb}\,, \tag{6-2}$$

with $\tau = 1$. The entries in the column for the response functions are the folding response function set (used to produce the exact signals) followed by the unfolding response function set (used to obtain the unfolded fluxes). The calibration and counting errors are listed next, and this is self-explanatory. The column titled Max Error contains the maximum percentage error, by bin, between the discretized flux and the unfolded flux. The column titled Ave Error contains the average percentage error, by bin, between the discretized flux and the unfolded flux. The final columns are the overall maximum and average absolute values of the error ratios of the unfolded fluxes for all the bins and trials, $|\varepsilon_{rel}|^{\max}$ and $|\varepsilon_{rel}|^{avg}$.

Table 4. A Summary of the Fundamental Results.

Test Case	Flux	Response Functions	ε^{CAL}	ε^{count}	Max Error	Ave Error	$\|\varepsilon_{rel}\|^{\max}$	$\|\varepsilon_{rel}\|^{avg}$
1	φ_{pl}	Cal 1/Ideal	0	0	213%	107%	---	---
2	φ_{mb}	Cal 1/Ideal	0	0	206%	116%	---	---
3	φ_{pl}	Cal 2/Ideal	0	0	777%	291%	---	---
4	φ_{mb}	Cal 2/Ideal	0	0	1903%	436%	---	---
5	φ_{pl}	Cal 2/Cal 1	0	0	190%	82%	---	---
6	φ_{mb}	Cal 2/Cal 1	0	0	562%	125%	---	---
7	φ_{pl}	Cal 1/Cal 1	0	0	23.1%	6.8%	---	---
8	φ_{pl}	Cal 2/Cal 2	0	0	13.9%	4.4%	---	---
9	φ_{pl}	Cal 1/Cal 1	5 %	1 %	~20%	~6%	---	---
10	φ_{pl}	Cal 2/Cal 2	5 %	1 %	32%	7%	0.276	0.069
11	φ_{pl}	Cal 1/Cal 1	20 %	1 %	~20%	~9%	---	---
12	φ_{pl}	Cal 2/Cal 2	20 %	1 %	35%	11%	0.297	0.109
13	φ_{pl}	Cal 1/Cal 1	20 %	5 %	~25%	~10%	---	---
14	φ_{pl}	Cal 2/Cal 2	20 %	5 %	306%	27%	1.209	0.235

Inferences from the Observations

Based on the experimentation presented in chapter five and summarized in table 3, inferences can be organized into four different categories. The first category pertains to approximating the set of response functions from the first calibration with an idealized set of response functions. Test cases 1 and 2 show this greatly and unnecessarily increases the errors introduced into the unfolded fluxes. The second category (test cases 3 and 4) examines the use of a set of idealized response functions as an approximation to the response functions extrapolated from the partial recalibration of the HEEF. This approximation introduces extremely large (and unnecessary) errors. The third category (test cases 5 and 6) uses the set of response functions derived from the first calibration to unfold data constructed using the second set of HEEF response functions. Using the original calibration data to unfold measurements when the measuring instrument is responding as we extrapolated from the partial recalibration introduces errors on the order of a factor of two to a factor of six, even before the introduction of counting and calibration error. The last category (tests 7 through 14) explores the propagation and growth of random calibration and counting errors. This error analysis includes both sets of HEEF response functions. The degree of coupling (resulting from the shape of the response functions) is the most important consideration for this category. In the absence of random errors, the second set of response functions performs better (than the first set) in calculating unfolded fluxes. This is a consequence of the second calibration response functions' smaller variation within energy bins, as compared to those of the first calibration. However, the strong degree of coupling in the second set has opened the door for more severe growth of random calibration and measurement errors as they propagate through the unfolding calculations. Thus, when counting and calibration error are added to the unfolding, the performance of the second set of response functions rapidly deteriorates while the performance of the first set remains fairly constant. The shape of the response functions is critical to the unfolding process, for it is the shape of the instrument sensitivities which drives the degree of coupling. In addition, for the flux

$$\varphi(E) = E^{-4} \tag{6-3}$$

the dominant source of error is the unavoidable counting errors in the low-signal channels.

Conclusions

If the response functions extrapolated from the partial recalibration of the HEEF represent the actual instrument sensitivities, than these response functions must be used for unfolding the high energy electron flux. Ignoring the need for an unfold (using a trivial unfold) could result in order-of-magnitude errors. (If the initial calibration were accurate, the errors introduced in this way would only be factor-of-two errors.) These response functions contain a large degree of coupling between successive instrument channels, and so errors from the instrument calibration and the fluxmeter signals must be minimized. The HEEF measurements already contain random counting errors. Perhaps data can be pooled to reduce them. In any event, however, calibration errors should be minimized.

The extrapolated second calibration response functions postulated in this study should not be considered conservative. It may well be that the actual HEEF response functions are even more strongly coupled and have more variation within energy bins. Unfortunately, trivial unfolding of the measurements, based on the currently available calibration data can produce plausible spectra that are used, but are seriously in error. Such errors cannot be estimated or bounded until the real response functions are accurately calibrated, over the full range of the instruments' response, and with good resolution. It may be that, once the response functions are known as accurately as possible, the unfolded flux spectra will be subject to substantial and unavoidable uncertainties. Is so, we need to be aware of that fact, so that poor results won't be given undue weight in space environment models.

Recommendations

As a result of the research done during this thesis, there are two recommendations to be made. First, a definitive recalibration of the HEEF must be completed before its measurements are trusted. Once this is accomplished, more robust unfolding techniques should be applied to the instrument measurements. One technique which may work well is a Backus-Gilbert formal regularization. An unfold based on a sound technique, in conjunction with a complete second calibration, will produce unfolded electron fluxes with a minimal amount of error, and perhaps more importantly, with knowable error bounds. The data gathered by CRRES are too valuable to the Department of Defense to lose, but the HEEF data can't be used safely without a definitive calibration and proper unfolding of high energy electron flux spectra.

Appendix A: The Fortran 90 Code

This appendix contains the Fortran 90 code used during the research stage of the thesis. Each numerical experiment generates output to the screen for analysis by the researcher, and so it is necessary to define the values which are displayed on the computer monitor. The first row in the display contains five entries. Starting on the left-hand side of the screen is a list of the instrument channels, and these values index the rest of the data in this row. The second set of numbers are the exact signals which correspond to y_i^E. The third component in this row is the vector of measured signals, y_i^M. The column titled Ideal Rspns contains the diagonal elements of the R_{ik}^E matrix. The next column, Calib'd Rspns, lists the diagonal elements of the R_{ik}^C matrix. The second row begins with a listing of the wide bins. Note that the number of instrument channels will always equal the number of wide bins. The next column entry is the exact flux, φ_k^E. The third column in the second row is the unfolded flux, φ_k^U. The next entry, titled Flux %Err, is defined as

$$\frac{\varphi_k^U - \varphi_k^E}{\varphi_k^E}. \tag{A-1}$$

If Flux %Err is negative, the unfolded flux is less than the exact (known) flux. The next column, DSig, is the difference between y_i^E and y_i^M. The final column, DRes, is the difference between R_{ik}^E and R_{ik}^C.

```
Program InstrumentUnfold

! This program allows the user to unfold a flux from a set of instrument signals.
! Initially, the user generates instrument signals from a known flux, and so he
! must select which flux and which response function will produce these signals.
! The response function matrix and the flux vector are used with the MatMul
! intrinsic function to generate the instrument signals. These signals and the
! response function picked by the user for unfolding are utilized in a Jacobi
! iterative scheme to produce an unfolded flux. In addition, the user has the
! option to add three types of error to the unfolding process. The first simulates
! counting error in the signal measurements. The second type models a systematic
! error in the calibration of the instrument. The final type represents an energy
! dependent error in the instrument calibration. These instrument calibration
! errors may take either an uniform or a gaussian distribution. In addition, the
! user may generate the random error manually (by inputting the seed), or he can
! let the machine generate the errors (in this case, the random error generating
! functions are seeded by a call of the system clock). Throughout the course of
! the program run, important information is written to data files which can be
! utilized in Mathematica, v2.2 notebook shells. This tracks information like
! response functions used, types of error added to the process, and flux values
! which can be plotted via BarChart in Mathematica.

Implicit None

        !!! REAL DECLARATIONS !!!

Real(8) temperature                    ! Temperature of the Maxwellian spectrum
Real(8) ErrorPercentSignal             ! Upper bound for user input signal error
Real(8) SysErrorPercentRespns          ! Upper bound for user input systematic
                                       ! response error
Real(8) BinErrorPercentRespns          ! Upper bound for user input energy dependent
                                       ! response error
Real(8) yMax                           ! The maximum value of the array for the
                                       ! exact signals measured by the instrument
Real(8) power                          ! Exponent value for the ActualFlux Function

        !!! INTEGER DECLARATIONS !!!

Integer, Parameter:: iMax = 10         ! Number of instrument channels
Integer RefineFactor                   ! Refinement factor for the narrow bins
Integer nw                             ! Number of wide bins in the unfold
Integer sChoice                        ! Case statement index for spectrum choice
Integer WriteChoice                    ! Case statement index for data write choice
Integer rFoldChoice                    ! Case statement index for folding
                                       ! response function
Integer rUnfoldChoice                  ! Case statement index for unfolding
                                       ! response function
Integer EbwChoice                      ! Case statement index for the choice of the
                                       ! Ebw array
```

```
Integer SignalErrorChoice              ! Case statement index for signal
                                       ! error choice
Integer ResponseSysErrorChoice         ! Case statement index for systematic
                                       ! response error choice
Integer ResponseBinErrorChoice         ! Case statement index for energy dependent
                                       ! response error choice
Integer ResSysErrorFormat              ! Case statement index for systematic
                                       ! response error format
IntegerResBinErrorFormat               ! Case statement index for energy dependent
                                       ! response error format
Integer jTotal                         ! Number of narrow energy bins
Integer i,j,k                          ! Counting indices
Integer Count1                         ! Assignment scalar for call of
                                       ! system clock (signal)
Integer Count2                         ! Assignment scalar for call of
                                       ! system clock (systematic)
Integer Count3                         ! Assignment scalar for call of
                                       ! system clock (energy dep)

        !!! INTEGER ARRAY DECLARATIONS !!!

Integer, Dimension(1) :: SeedInput1    ! Array for user input random seed (signal)
Integer, Dimension(1) :: SeedInput2    ! Array for user input random seed
                                       ! (systematic)
Integer, Dimension(1) :: SeedInput3    ! Array for user input random seed
                                       ! (energy dependent)
Integer, Dimension(1) :: Seed1         ! Array for random seed (signal)
Integer, Dimension(1) :: Seed2         ! Array for random seed (systematic)
Integer, Dimension(1) :: Seed3         ! Array for random seed (energy dep)
Integer, Allocatable :: jMax(:)        ! Index of the highest energy narrow bin in
                                       ! the wide bin
Integer, Allocatable :: nn(:)          ! Number of narrow bins in a wide bin

        !!! REAL ARRAY DECLARATIONS !!!

Real(8), Allocatable :: R1(:)          ! Array for random function values
                                       ! (1D, uniform)
Real(8), Allocatable :: R1A(:)         ! Array for random function values
                                       ! 1D, uniform systematic response error)
Real(8), Allocatable :: R2(:,:)        ! Array for random function values
                                       ! (2D, uniform)
Real(8), Allocatable :: R2A(:,:)       ! Array for random function values (2D,
                                       ! uniform systematic response error)
Real(8), Allocatable :: R3(:,:,:)      ! Array for random function values
                                       ! (2D, uniform)
```

```
Real(8), Allocatable :: G1(:)                  ! Gaussian random number
Real(8), Allocatable :: G1A(:)                 ! Gaussian random number (scalar, gaussian,
                                               ! systematic response error)
Real(8), Allocatable :: G2(:,:)                ! Array for random function values
                                               ! (1D, gaussian)
Real(8), Allocatable :: RnFold(:,:)            ! Response functions for narrow bin folding
Real(8), Allocatable :: RnUnfold(:,:)          ! Response functions for narrow bin unfolding
Real(8), Allocatable :: RwUnfoldExact(:,:)     ! Exact response functions for
                                               ! wide bin unfolding
Real(8), Allocatable :: RwUnfoldBinError(:,:)! Energy Dependent Error in the response for
                                               ! wide bin unfolding
Real(8), Allocatable :: RwUnfoldSysError(:,:)! Systematic Error in the response for wide
                                               ! bin unfolding
Real(8), Allocatable :: RwUnfoldCalib(:,:)     ! Simulated calibrated response function for
                                               ! the wide bin
Real(8), Allocatable :: nFlux(:)               ! Flux spectrum for the narrow bins
Real(8), Allocatable :: wFlux(:)               ! Flux spectrum for the wide bins
Real(8), Allocatable :: dEn(:)                 ! Width of the narrow bins
Real(8), Allocatable :: FluxUnfolded(:)        ! Output for the linear solve subprogram
Real(8), Allocatable :: DeltaWideFlux(:)       ! Absolute value of the difference between
                                               ! the wide flux and the unfolded flux
Real(8), Allocatable :: FluxErrorPcnt(:)       ! Percentage error based on the difference
                                               ! between the wide flux and the unfolded flux
Real(8), Allocatable :: DeltaSig(:)            ! Array containing the absolute value of the
                                               ! the difference between exact and
                                               ! measured signal
Real(8), Allocatable :: DeltaRes(:,:)          ! Array containing the abs value of the
                                               ! difference between exact and calib response
Real(8), Allocatable :: yExact(:)              ! Exact instrument signals
Real(8), Allocatable :: yMeasured(:)           ! Measured instrument signals
Real(8), Allocatable :: yDelta(:)              ! Error in the measured instrument signal
Real(8), Allocatable :: Ebw(:)                 ! Energy at upper boundary of a wide bin
Real(8), Allocatable :: Ebc(:)                 ! Energy at the center of a wide bin
Real(8), Allocatable :: dEnw(:)                ! Width of the narrow bin within a wide bin
Real(8), Allocatable :: dEw(:)                 ! Width of a wide bin
Real(8), Allocatable :: En(:)                  ! Energy of the particles


        !!! INTERFACE BLOCK !!!

! The following section of code contains the interface block for the seven
! flux generation functions, the six response functions, and the Jacobi
! linear solve subroutine.

Interface
```

```
Real(8) Function Maxwellian(e,tau)
    Real(8) e                        ! energy of particles
    Real(8) tau                      ! fundamental temperature
End Function Maxwellian

Real(8) Function ExponentialD(e)
    Real(8) e                        ! energy of the particle
End Function ExponentialD

Real(8) Function LinearD(e)
    Real(8) e                        ! energy of the particle
End Function LinearD

Real(8) Function ConExponD(e)
    Real(8) e                        ! energy of the particle
End Function ConExponD

Real(8) Function ConLinD(e)
    Real(8) e                        ! energy of the particle
End Function ConLinD

Real(8) Function Heaviside(e)
    Real(8) e                        ! energy of the particle
End Function Heaviside

Real(8) Function ActualFlux(e,p)
    Real(8) e                        ! energy of the particle
    Real(8) p                        ! value for the exponent
End Function ActualFlux

Real(8) Function ResponseA(i,e,wbeb)
    Integer i                        ! index counter (instrument channel)
    Real(8) e                        ! energy of the particle
    Real(8), Dimension(0:):: wbeb    ! wide bin energy boundary
End Function ResponseA

Real(8) Function ResponseB(i,e,wbeb)
    Integer i                        ! index counter (instrument channel)
    Real(8) e                        ! energy of the particle
    Real(8), Dimension(0:):: wbeb    ! wide bin energy boundary
End Function ResponseB

Real(8) Function ResponseC(i,e,wbeb)
    Integer i                        ! index counter (instrument channel)
    Real(8) e                          ! energy of the particle
    Real(8), Dimension(0:):: wbeb      ! wide bin energy boundary
End Function ResponseC

Real(8) Function ResponseD(i,e,wbeb)
    Integer i                          ! index counter (instrument channel)
    Real(8) e                          ! energy of the particle
    Real(8), Dimension(0:):: wbeb      ! wide bin energy boundary
End Function ResponseD

Real(8) Function HEEFresOne(i,j,e,jTotalpass)
    Integer i                          ! index counter (instrument channel)
    Integer j                          ! index counter (narrow bin channel)
    Real(8) e                          ! energy of the particle
    Integer jTotalpass                 ! total number of narrow bins
End Function HEEFresOne

Real(8) Function HEEFresTwo(i,e)
    Integer, Intent(IN):: i            ! index counter (instrument channel)
    Real(8), Intent(IN):: e            ! energy of the particle
End Function HEEFresTwo

Subroutine Jacobi(x,A,b)
    Real(8), Intent(Out):: x(:)        ! This array is the iterated solution
                                       ! the Jacobi subroutine returns (flux)
    Real(8), Intent(In):: A(:,:)       ! The array for the response functions
    Real(8), Intent(In):: b(:)         ! The array for the instrument signals
End Subroutine

End Interface

        !!! INTRODUCTION OF THE CODE FOR THE USER  !!!

! The block of text below provides the user with a basic description of
! what this program does. It is shown at the beginning of each program
! run.

Print*, "Program InstrumentUnfold:"
Print*
Print*, "This program allows the user to unfold a flux from a set of instrument"
Print*, "signals. The simulated signals are generated from a known flux where the"
Print*, "user selects which flux and which set of response functions produce these"
Print*, "signals. The flux and the set of response functions are used with the"
Print*, "MATMUL intrinsic function to generate exact instrument signals. The"
Print*, "signals and a response function picked by the user for unfolding are"
```

```
Print*, "utilized in a Jacobi iterative scheme to unfold the flux. The user may"
Print*, "add three types of error to the unfolding process. The first simulates"
Print*, "counting error in the signal measurements. The second models a systematic"
Print*, "error in the calibration of the instrument. The final type represents an"
Print*, "energy dependent error in the instrument response. The program output is"
Print*, "displayed on the screen and written to data files for later use."
Print*

nw = iMax      ! Required for the direct inversion done by the Jacobi Subroutine

        !!! SELECTION BLOCK FOR WRITING DATA TO FILES !!!

! The user now has the option of writing data to files for later use in the
! Mathematica, v2.2 software. The path set up for the data save is machine
! specific, so it is suggested that the user NOT write any data to files unless
! this path has been altered for the specific machine.

Print*, " Many of the responses made to the series of questions which"
Print*, " follow can be written to files stored in the Mathematica"
Print*, " directory. These data files can then be examined at a later"
Print*, " time. Please select which option you wish to use for this"
Print*, " data run (choice must be an integer)."
Print*, "  1 = Write data to files in the c:\wnmath22 directory "
Print*, "  2 = Do not write any data to files "
Print*
Write(*,'(A)',Advance='NO') "   Choice = "
Read(*,*) WriteChoice
Print*

        !!! FLUX FORM CHOICE !!!

! The section of code below asks the user to input the form of the energy
! spectrum required for the folding process. There are seven different
! selections. Note that the Maxwellian form for the flux requires the
! user to input a temperature (in MeVs) for the location of the peak of the
! Maxwellian distribution function. The peak will be located at one half
! the value of the temperature. Choice seven is a form of the spectrum
! based on actual measurements made by the instrument. The information
! required to formulate this choice was obtained from the sponsor, and it
! requires input from the user in the form of a negative power with
! which to raise the energy. Components of the user input are written to
! data files for use with Mathematica.

Print*, " Enter your selection for the functional form"
Print*, " of the energy spectrum (must be an integer)."
```

```
Print*, "  1 = Maxwellian "
Print*, "  2 = Exponential Decrease    "
Print*, "  3 = Linear Decrease "
Print*, "  4 = Constant Followed by Exponential Decrease "
Print*, "  5 = Constant Followed by Linear Decrease "
Print*, "  6 = Heaviside (constant than a factor of 100 decrease)"
Print*, "  7 = Particle Energy Raised to the Negative 'p' "
Print*
Write(*,'(A)',Advance='NO')  "   Choice = "
Read(*,*) sChoice
Print*

! The code below is constructed for writing data files

If (WriteChoice .EQ. 1) Then
 Select Case (sChoice)
   Case (1)
     Open (Unit = 3, File = 'C:\wnmath22\sChoice.dat')
     Write (3,*) "The form of the spectrum for this data run is Maxwellian."
     Close (3)
   Case (2)
     Open (Unit = 3, File = 'C:\wnmath22\sChoice.dat')
     Write (3,*) "The form of the spectrum for this data run&
                  is an exponential decrease."
     Close (3)
   Case (3)
     Open (Unit = 3, File = 'C:\wnmath22\sChoice.dat')
     Write (3,*) "The form of the spectrum for this data run&
                  is a linear decrease."
     Close (3)
   Case (4)
     Open (Unit = 3, File = 'C:\wnmath22\sChoice.dat')
     Write (3,*) "The form of the spectrum for this data run&
                  is a constant followed by an exponential decrease."
     Close (3)
   Case (5)
     Open (Unit = 3, File = 'C:\wnmath22\sChoice.dat')
     Write (3,*) "The form of the spectrum for this data run&
                  is a constant followed by a linear decrease."
     Close (3)
   Case (6)
     Open (Unit = 3, File = 'C:\wnmath22\sChoice.dat')
     Write (3,*) "The form of the spectrum for this data run&
                  is a Heaviside (constant then a factor of 100 decrease)."
     Close (3)
   Case (7)
      Open (Unit = 3, File = 'C:\wnmath22\sChoice.dat')
      Write (3,*) "The form of the spectrum for this data run&
                  is particle energy raised to a negative power."
      Close (3)
  End Select
End If

        !!! TEMPERATURE INPUT FOR MAXWELLIAN FLUX !!!

! If the user selects a flux with a Maxwellian form, the section of code
! below will ask the user to input a temperature which locates the peak
! of the Maxwellian distribution function. The peak will be located at one
! half the value of the input temperature. This information is written
! to a data file for later use.

If (sChoice .EQ. 1) Then
   Print*, " Enter temperature of Maxwellian Spectrum (MeV): "
   Print*
   Write(*,'(A)',Advance='NO')      "  Temperature = "
   Read(*,*) temperature
   Print*

   ! The code below is constructed for writing data files

   If (WriteChoice .EQ. 1) Then
      Open (Unit = 4, File = 'C:\wnmath22\temp.dat')
      Write (4,*) "The location of the peak (in MeV) of the Maxwellian&
                  spectrum is", temperature/2 ,"."
      Close (4)
   End If
End If

        !!! POWER INPUT FOR 'ENERGY TO A POWER' FLUX !!!

! If the user selects a flux of the form denoted by the sponsor at Phillips
! Lab, the section of code below will ask the user to input a power for
! which to raise the energy value passed to the function. Although the
! power is negative, the user is prompted for a positive input. The sign
! is changed in the code. This information is written to a data file
! for later use.

If (sChoice .EQ. 7) Then
   Print*, " Enter the value for the power (positive number) "
   Print*, " which determines the rate of decrease for the flux. "
```

```
    Print*, " This can be a real number.  The larger the number, "
    Print*, " the greater the rate of decrease. "
    Print*
    Write(*,'(A)',Advance='NO')        "   Power = "
    Read(*,*) power
    Print*

    ! The code below is constructed to write data to files

    If (WriteChoice .EQ. 1) Then
        Open (Unit = 5, File = 'C:\wnmath22\power.dat')
        Write (5,*) "The power to which the energy is raised for&
                        the form of the spectrum is", -power ,"."
        Close (5)
    End If
End If

          !!!  FOLDING RESPONSE CHOICE  !!!

! The following section of code prompts the user to input the form of the
! response function required for folding with the flux in order to produce
! a set of instrument signals.  There are six different options from which
! to choose.  Information is written to a data file for later use.

Print*, " Enter your selection for the functional form"
Print*, " of the folding response function (must be an integer)."
Print*, "  1 = Idealized Response for the 1st. Calibration "
Print*, "  2 = Idealized Response (Heaviside) for the 2nd. Calibration "
Print*, "  3 = Refined Response for the 1st. Calibration "
Print*, "  4 = Refined Response for the 2nd. Calibration"
Print*, "  5 = First Calibration of the HEEF "
Print*, "  6 = Second Calibration of the HEEF "
Print*
Write(*,'(A)',Advance='NO')  "   Choice = "
Read(*,*) rFoldChoice
Print*

! The code below is designed to write data to files

If (WriteChoice .EQ. 1) Then
  Select Case (rFoldChoice)
   Case (1)
       Open (Unit = 6, File = 'C:\wnmath22\rFoldChc.dat')
       Write (6,*) "The choice for the folding response function is&
                        an ideal response."
```

```
      Close (6)
    Case (2)
      Open (Unit = 6, File = 'C:\wnmath22\rFoldChc.dat')
      Write (6,*) "The choice for the folding response function is&
                  a Heaviside response."
      Close (6)
    Case (3)
      Open (Unit = 6, File = 'C:\wnmath22\rFoldChc.dat')
      Write (6,*) "The choice for the folding response function is&
                  an equal tail size response."
      Close (6)
    Case (4)
      Open (Unit = 6, File = 'C:\wnmath22\rFoldChc.dat')
      Write (6,*) "The choice for the folding response function is&
                  a large energy tail response."
      Close (6)
    Case (5)
      Open (Unit = 6, File = 'C:\wnmath22\rFoldChc.dat')
      Write (6,*) "The choice for the folding response function is&
                  the first calibration of the HEEF."
      Close (6)
    Case (6)
      Open (Unit = 6, File = 'C:\wnmath22\rFoldChc.dat')
      Write (6,*) "The choice for the folding response function is&
                  the second calibration of the HEEF."
      Close (6)
  End Select
End If

        !!! UNFOLDING RESPONSE CHOICE !!!

! The following section of code prompts the user to input the form of the
! response function required for unfolding the flux from the instrument
! signals. There are six different options from which to choose. This
! information is written to a file for later user.

Print*, " Enter your selection for the functional form"
Print*, " of the unfolding response function (must be an integer)."
Print*, "  1 = Idealized Response for the 1st. Calibration "
Print*, "  2 = Idealized Response (Heaviside) for the 2nd. Calibration "
Print*, "  3 = Refined Response for the 1st. Calibration "
Print*, "  4 = Refined Response for the 2nd. Calibration"
Print*, "  5 = First Calibration of the HEEF "
Print*, "  6 = Second Calibration of the HEEF "
Print*
```

```
Write(*,'(A)',Advance='NO')  "   Choice = "
Read(*,*) rUnFoldChoice
Print*

! The code below is designed to write data to files

If (WriteChoice .EQ. 1) Then
  Select Case (rUnFoldChoice)
   Case (1)
      Open (Unit = 7, File = 'C:\wnmath22\rUnFldCh.dat')
      Write (7,*) "The choice for the unfolding response function is&
                   an ideal response."
      Close (7)
   Case (2)
      Open (Unit = 7, File = 'C:\wnmath22\rUnFldCh.dat')
      Write (7,*) "The choice for the unfolding response function is&
                   a Heaviside response."
      Close (7)
   Case (3)
      Open (Unit = 7, File = 'C:\wnmath22\rUnFldCh.dat')
      Write (7,*) "The choice for the unfolding response function is&
                   an equal tail size response."
      Close (7)
   Case (4)
      Open (Unit = 7, File = 'C:\wnmath22\rUnFldCh.dat')
      Write (7,*) "The choice for the unfolding response function is&
                   a large energy tail response."
      Close (7)
   Case (5)
      Open (Unit = 7, File = 'C:\wnmath22\rUnFldCh.dat')
      Write (7,*) "The choice for the unfolding response function is&
                   the first calibration of the HEEF."
      Close (7)
   Case (6)
      Open (Unit = 7, File = 'C:\wnmath22\rUnFldCh.dat')
      Write (7,*) "The choice for the unfolding response function is&
                   the second calibration of the HEEF."
      Close (7)
  End Select
End If


        !!!  Ebw ARRAY CHOICE  !!!

! The following section of code prompts the user to select the desired form
! for the array which represents the energy values at the boundaries of all of
! the wide bins (Ebw). Note that the choice for the unfolding response
! function will drive this selection. The Ebw array must correspond to the
! response function the user wishes to use for unfolding. This information
! is written to a data file for later use.

Print*, " Enter your selection for the format of the array which"
Print*, " represents the values for the energies at the wide bin"
Print*, " boundaries (Ebw). Note that this selection must correspond"
Print*, " to the unfolding response function (must be an integer)."
Print*, "  1 = Equally Spaced Energy Boundaries (except for last bin) "
Print*, "  2 = Wide Bin Boundaries corresponding to the first "
Print*, "      calibration of the HEEF "
Print*, "  3 = Wide Bin Boundaries corresponding to the second "
Print*, "      calibration of the HEEF "
Print*
Write(*,'(A)',Advance='NO') " Choice = "
Read(*,*) EbwChoice
Print*

! The section of code below writes data to files.

If (WriteChoice .EQ. 1) Then
  Select Case (EbwChoice)
   Case (1)
      Open (Unit = 8, File = 'C:\wnmath22\Ebwchc.dat')
      Write (8,*) "The wide bin boundries correspond to equally spaced&
                  energy values."
      Close (8)
   Case (2)
      Open (Unit = 8, File = 'C:\wnmath22\Ebwchc.dat')
      Write (8,*) "The wide bin boundries correspond to energy values&
                  from the first instrument calibration."
      Close (8)
   Case (3)
      Open (Unit = 8, File = 'C:\wnmath22\Ebwchc.dat')
      Write (8,*) "The wide bin boundries correspond to energy values&
                  from the second instrument calibration."
      Close (8)
   End Select
End If

        !!! REFINEMENT FACTOR CHOICE !!!

! The following section of code prompts the user to input an INTEGER for use
! as an refinement factor.  This refinement factor will dictate the number of
! narrow bins contained within a wide bin.  Do not use a value less than 1.

Print*, " Enter your selection for the refinement factor for the"
Print*, " narrow bins.  This refinement factor will dictate how many"
Print*, " narrow bins are contained within a given wide bin.  A factor"
Print*, " selection of one equates to the standard option.  Higher "
Print*, " numbers will increase the number of narrow bins per wide bin. "
Print*, " Do not select a factor less than one, and the selection must"
Print*, " be an integer."
Print*
Write(*,'(A)',Advance='NO') " Refinement Factor = "
Read(*,*) RefineFactor
Print*

          !!! SIGNAL ERROR !!!

! Here the user inputs the upper bound on the gaussian
! distributed counting error which is added to the signal measurement.
! This number should be entered as a percentage (0.5% should
! be entered as 0.5).  Be sure to note the format of the error calculation
! in the section of code which calculates yDelta, because an input of 0.5
! does not mean the signals will have an error of 0.5%.  This is a scaling
! factor involved in the error generation.

Print*, " Enter the percentage (real) for the upper bound"
Print*, " of the signal counting error (user may enter 0)"
Print*
Write(*,'(A)',Advance='NO')  "   Error Percentage for Upper Limit = "
Read(*,*) ErrorPercentSignal
Print*

        !!! USER OR MACHINE OPTION FOR SIGNAL ERROR !!!

! The section of code below prompts the user to make a choice for the type
! of random counting error added to the exact signal.  The user can either
! seed the random number generator himself, or he can let the machine do it
! by use of the system clock function.  This information is written to a
! data file for later use.

If (ErrorPercentSignal > 0) Then
   Print*, " Enter your choice for the generation of error"
   Print*, " for the signal measurements (must be an integer)."
   Print*, "  1 for a user generated random error"
   Print*, "  2 for a machine generated random error"
```

```
    Print*
    Write(*,'(A)',Advance='NO')          "  Choice = "
    Read(*,*) SignalErrorChoice
    Print*

          ! The code below writes data to files.

   If (WriteChoice .EQ. 1) Then
    Open (Unit = 15, File = 'C:\wnmath22\SigErPnt.dat')
    Write (15,*) "The error percentage for the signal measurements&
                   is", ErrorPercentSignal, "."
    Close(15)
    Select Case (SignalErrorChoice)
       Case (1)
          Open (Unit = 9, File = 'C:\wnmath22\SigErTyp.dat')
          Write (9,*) "The errors in the signal measurements are&
                       generated by the user."
          Close (9)
       Case (2)
          Open (Unit = 9, File = 'C:\wnmath22\SigErTyp.dat')
          Write (9,*) "The errors in the signal measurements are&
                       generated by the machine."
          Close (9)
    End Select
   End If

  Else
          ! The code below writes data to files.

    If (WriteChoice .EQ. 1) Then
       Open (Unit = 9, File = 'C:\wnmath22\SigErTyp.dat')
       Write (9,*) "There are no errors in the signal measurements."
       Close (9)
    End If

  End If

          !!!  SYSTEMATIC ERROR FOR THE RESPONSE  !!!

  ! Here the user inputs the upper bound on the uniformly or gaussian
  ! distributed systematic error which is added to the response function.
  ! This number should be entered as a percentage (20% should be
  ! entered as 20).

  Print*, " Enter the percentage (real) for the upper bound of the"
```

```
Print*, " systematic response function error (user may enter 0)"
Print*
Write(*,'(A)',Advance='NO')  "   Error Percentage for Upper Limit = "
Read(*,*) SysErrorPercentRespns
Print*

        !!! USER OR MACHINE OPTION FOR SYSTEMATIC ERROR !!!

! The section of code below prompts the user to make a choice for the type
! of random error added to the response function. The user can either seed
! the random number generator himself, or he can let the machine do it by
! use of the system clock function.

If (SysErrorPercentRespns > 0) Then
   Print*, " Enter your choice for the generation of systematic"
   Print*, " error for the response function (must be an integer)."
   Print*, "  1 for a user generated random error"
   Print*, "  2 for a machine generated random error"
   Print*

   ! The code below writes data to files

   Write(*,'(A)',Advance='NO')        "   Choice = "
   Read(*,*) ResponseSysErrorChoice
   Print*
   If (WriteChoice .EQ. 1) Then
     Open (Unit = 16, File = 'C:\wnmath22\SysErPtR.dat')
     Write (16,*) "The systematic error percentage for the calibration&
        is", SysErrorPercentRespns, "."
     Close(16)

     Select Case (ResponseSysErrorChoice)
        Case (1)
           Open (Unit = 11, File = 'C:\wnmath22\ReSErTyp.dat')
           Write (11,*) "The systematic errors in the instrument&
                         calibration are generated by the user."
           Close (11)
        Case (2)
           Open (Unit = 11, File = 'C:\wnmath22\ReSErTyp.dat')
           Write (11,*) "The systematic errors in the instrument&
                         calibration are generated by the machine."
           Close (11)
        End Select
  End If
```

```
Else
        ! The section of code below writes data to files

   If (WriteChoice .EQ. 1) Then
       Open (Unit = 11, File = 'C:\wnmath22\ReSErTyp.dat')
       Write (11,*) "There are no systematic errors in the calibration."
       Close (11)
   End If

End If

        !!!  SYSTEMATIC ERROR FORMAT--GAUSSIAN OR UNIFORM  !!!

! In the following text block the user is asked to select between a
! gaussian or uniformly distributed systematic error.  This allows the
! user to simulate different types of systematic error which may occur
! with a calibration.

If (SysErrorPercentRespns > 0) Then
   Print*, " Enter your choice for the form of the"
   Print*, " systematic response error (must be an integer)."
   Print*, "  1 for an uniformly distributed error"
   Print*, "  2 for a gaussian distributed error"
   Print*
   Write(*,'(A)',Advance='NO')         "   Choice = "
   Read(*,*) ResSysErrorFormat
   Print*

        ! The code below writes data to files

   If (WriteChoice .EQ. 1) Then
     Select Case (ResSysErrorFormat)
        Case (1)
            Open (Unit = 12, File = 'C:\wnmath22\ReSErFmt.dat')
            Write (12,*) "The systematic errors in the calibration&
                          area uniformily distributed."
            Close (12)
        Case (2)
            Open (Unit = 12, File = 'C:\wnmath22\ReSErFmt.dat')
            Write (12,*) "The systematic errors in the calibration&
                          are distributed in a gaussian manner."
            Close (12)
     End Select
   End If
End If

        !!! ENERGY DEPENDENT ERROR FOR THE RESPONSE  !!!

! Here the user inputs the upper bound on an energy dependent uniformly or
! gaussian distributed error which is added to the response function.  This
! number should be entered as a percentage (20% should be entered as 20).

Print*, " Enter the percentage (real) for the upper bound of the"
Print*, " energy dependent response function error (user may enter 0)"
Print*
Write(*,'(A)',Advance='NO')  "   Error Percentage for Upper Limit = "
Read(*,*) BinErrorPercentRespns
Print*

        !!! USER OR MACHINE OPTION FOR ENERGY DEPDNT ERROR  !!!

! The section of code below prompts the user to make a choice for the type
! of energy dependent random error added to the response function.  The user
! can either seed the random number generator himself, or he can let the
! machine do it by use of the system clock function.

If (BinErrorPercentRespns > 0) Then
   Print*, " Enter your choice for the generation of energy dependent"
   Print*, " error for the response function (must be an integer)."
   Print*, "  1 for a user generated random error"
   Print*, "  2 for a machine generated random error"
   Print*
   Write(*,'(A)',Advance='NO')        "   Choice = "
   Read(*,*) ResponseBinErrorChoice
   Print*

        ! The code below writes data to files

  If (WriteChoice .EQ. 1) Then
    Open (Unit = 17, File = 'C:\wnmath22\BinErPtR.dat')
    Write (17,*) "The energy dependent error percentage for the&
                  calibration is", BinErrorPercentRespns, "."
    Close(17)

    Select Case (ResponseBinErrorChoice)
        Case (1)
            Open (Unit = 13, File = 'C:\wnmath22\ReBErTyp.dat')
            Write (13,*) "The energy dependent errors in the instrument&
                          calibration are generated by the user."
            Close (13)
        Case (2)
            Open (Unit = 13, File = 'C:\wnmath22\ReBErTyp.dat')
            Write (13,*) "The energy dependent errors in the instrument&
                                    calibration are generated by the machine."
            Close (13)
    End Select
  End If

Else
        ! The code below writes data to files

  If (WriteChoice .EQ. 1) Then
    Open (Unit = 13, File = 'C:\wnmath22\ReBErTyp.dat')
    Write (13,*) "There are no energy dependent errors in the calibration."
    Close (13)
  End If

End If

        !!!  ENERGY DEPDNT ERROR FORMAT--GAUSSIAN OR UNIFORM  !!!

! In the following text block the user is asked to select between a
! gaussian or uniformly distributed energy dependent error.  This
! allows the user to simulate different types of energy dependent
! error which may occur with a calibration.

If (BinErrorPercentRespns > 0) Then
   Print*, " Enter your choice for the form of the energy"
   Print*, " dependent response error (must be an integer)."
   Print*, "  1 for an uniformly distributed error"
   Print*, "  2 for a gaussian distributed error"
   Print*
   Write(*,'(A)',Advance='NO')        "   Choice = "
   Read(*,*) ResBinErrorFormat
   Print*

        ! The code below writes data to files

  If (WriteChoice .EQ. 1) Then
    Select Case (ResBinErrorFormat)
        Case (1)
            Open (Unit = 14, File = 'C:\wnmath22\ReBErFmt.dat')
            Write (14,*) "The energy dependent errors in the calibration&
                              are uniformily distributed."
            Close (14)
        Case (2)
            Open (Unit = 14, File = 'C:\wnmath22\ReBErFmt.dat')
            Write (14,*) "The energy dependent errors in the calibration&
                          are distributed in a gaussian manner."
            Close (14)
    End Select
  End If

End If

        !!! GATHER THE RANDOM SEEDS  !!!

! The code which follows will prompt the user for a random seed input in
! the case where the user wishes to generate random error in the signal
! measurements (as opposed to letting the machine generate random error
! by calling the system clock).

If (SignalErrorChoice==1) Then
   Print*, " Enter your integer input for the random seed"
   Print*, " (counting error in the signal)."
   Print*
   Write(*,'(A)',Advance='NO')        "   Seed = "
   Read(*,*) SeedInput1
   Print*

        ! The code below writes data to files

   If (WriteChoice .EQ. 1) Then
      Open (Unit = 18, File = 'C:\wnmath22\Seed1.dat')
      Write (18,*) "The signal counting error seed is", SeedInput1 ,"."
      Close(18)
   End If

End If

! The code which follows will prompt the user for a random seed input in
! the case where the user wishes to generate random error in the
! systematic response (as opposed to letting the machine generate random
! error by calling the system clock).

If (ResponseSysErrorChoice==1) Then
   Print*, " Enter your integer input for the random seed"
   Print*, " (systematic error in the calibration)."
   Print*
```

```
    Write(*,'(A)',Advance='NO')       "  Seed = "
    Read(*,*) SeedInput2
    Print*

    ! The code below writes data to files

    If (WriteChoice .EQ. 1) Then
      Open (Unit = 19, File = 'C:\wnmath22\Seed2.dat')
      Write (19,*) "The systematic response error seed is", SeedInput2 ,"."
      Close(19)
    End If

  End If

  ! The code which follows will prompt the user for a random seed input in
  ! the case where the user wishes to generate random error in the energy
  ! dependent response (as opposed to letting the machine generate random
  ! error by calling the system clock).

  If (ResponseBinErrorChoice==1) Then
    Print*, " Enter your integer input for the random seed"
    Print*, " (energy dependent error in the calibration)."
    Print*
    Write(*,'(A)',Advance='NO')       "  Seed = "
    Read(*,*) SeedInput3
    Print*

    ! The code below writes data to files

    If (WriteChoice .EQ. 1) Then
      Open (Unit = 20, File = 'C:\wnmath22\Seed3.dat')
      Write (20,*) "The energy dependent response error seed is", SeedInput3 ,"."
      Close(20)
    End If

  End If

  ! The required inputs from the user have been gathered.  The main body of
  ! code follows.  This is where the actual calculations for signal, flux,
  ! and instrument response are generated.


        !!!  MAIN BODY OF THE CODE  !!!
```

```
! Initial memory allocation and value assignments...

 Allocate(jMax(0:nw), nn(1:nw))

 jMax(0) = 0

 ! The section of code below assigns values to the array which represents
 ! the number of narrow bins within a wide bin. By using the refinement
 ! factor, the user can increase the number of narrow bins in each wide bin.

 nn = (/100, 100, 100, 100, 100,  &
         100, 200, 200, 400, 400/)

 nn = nn * RefineFactor

 ! The following do loop assigns values to the array jMax. This is the
 ! number of the last narrow energy bin within a given wide bin.

 Do k=1,nw
   jMax(k) = jMax(k-1) + nn(k)
 End Do

 ! jTotal is the total number of narrow bins contained within all of the
 ! wide bins.

 jTotal = jMax(nw)

 ! The section of code below allocates memory storage space for the majority
 ! of arrays used in this program.

 Allocate (RnFold(iMax,jTotal), nFlux(jTotal), wFlux(nw))
 Allocate (Ebw(0:nw), Ebc(nw), dEnw(nw), dEw(nw), dEn(jTotal), En(jTotal))
 Allocate (RnUnfold(iMax,jTotal), RwUnfoldExact(iMax,nw), RwUnfoldCalib(iMax,nw))
 Allocate (RwUnfoldBinError(iMax,nw), RwUnfoldSysError(iMax,nw))

 ! Ebw is the array which contains the energy values (in MeVs) for the
 ! boundary of each wide bin. The user has three options to choose from,
 ! and the selection is driven by the choice of the unfolding response
 ! function. Case 2 and Case 3 correspond to the instrument responses

 Select Case (EbwChoice)
   Case(1)
     Do k=0,nw-1
       Ebw(k) = 1.0 + k
     End Do
```

```
      Ebw(nw) = Ebw(nw-1)+2
   Case(2)
      Ebw = (/1.04, 1.56, 2.085, 2.581429, 3.042857, 3.557143, &
                4.057977, 5.024013, 6.090244, 8.066304, 10.0337/)
   Case(3)
      Ebw = (/1.15, 1.51, 1.85, 2.54, 3.025, 3.54, &
                4.205, 5.15, 6.66, 8.55, 10.03/)
End Select

! The block of code below calculates the energy value (in MeV) at center
! of each wide bin.

Do k=1,nw
    Ebc(k) = (Ebw(k-1) + Ebw(k))/2
End Do

! dEw is the array which contains the energy width of the wide bins.  It is
! calculated by taking the difference between successive energy boundary
! values.

dEw(1:nw) = Ebw(1:nw) - Ebw(0:nw-1)

! dEnw is the width of the narrow bins contained within a respective wide
! bin.  It is calculated by dividing the wide bin width by the number of
! narrow bins within that wide bin.

dEnw = dEw/nn

! The nested do loop below generates two arrays which are used in the
! generation of narrow bin response functions for both folding and
! unfolding.  The first step (calculating dEn) ensures all the narrow
! energy bins within a wide bin have the same width.  This is required
! for the two loops which follow this one.  The second step (calculating
! En) generates an average energy value for the middle of each narrow bin.
! This quantity is used by the functions which calculate values for both
! the narrow bin response functions and the narrow bin fluxes.

Do k=1,nw
    Do j=jMax(k-1)+1,jMax(k)
        dEn(j) = dEnw(k)
        En(j) = Ebw(k-1) + dEw(k)*(j-jMax(k-1)-0.5)/nn(k)
    End Do
End Do

        !!!  THE NARROW BIN FOLDING RESPONSE  !!!
```

```
! The nested do loop which follows generates an array which contains
! values for the narrow bin response function used for folding. Note
! that use of a case statement allows the user to select which of six
! narrow bin response functions he wishes to use for folding.

Do i=1,iMax
   Do j=1,jTotal
      Select Case (rFoldChoice)
         Case (1)
            RnFold(i,j) = ResponseA(i,En(j),Ebw)*dEn(j)
         Case (2)
            RnFold(i,j) = ResponseB(i,En(j),Ebw)*dEn(j)
         Case (3)
            RnFold(i,j) = ResponseC(i,En(j),Ebw)*dEn(j)
         Case (4)
            RnFold(i,j) = ResponseD(i,En(j),Ebw)*dEn(j)
         Case (5)
            RnFold(i,j) = HEEFresOne(i,j,En(j),jTotal)*dEn(j)
         Case (6)
            RnFold(i,j) = HEEFresTwo(i,En(j))*dEn(j)
      End Select
   End Do
End Do


        !!! THE NARROW BIN UNFOLDING RESPONSE !!!

! The nested do loop which follows generates an array which contains
! values for the narrow bin response function used for unfolding. Note
! that use of a case statement allows the user to select which of six
! narrow bin response functions he wishes to use for unfolding.

Do i=1,iMax
   Do j=1,jTotal
      Select Case (rUnfoldChoice)
         Case (1)
            RnUnfold(i,j) = ResponseA(i,En(j),Ebw)*dEn(j)
         Case (2)
            RnUnfold(i,j) = ResponseB(i,En(j),Ebw)*dEn(j)
         Case (3)
            RnUnfold(i,j) = ResponseC(i,En(j),Ebw)*dEn(j)
         Case (4)
            RnUnfold(i,j) = ResponseD(i,En(j),Ebw)*dEn(j)
         Case (5)
```

```
            RnUnfold(i,j) = HEEFresOne(i,j,En(j),jTotal)*dEn(j)
         Case (6)
            RnUnfold(i,j) = HEEFresTwo(i,En(j))*dEn(j)
      End Select
   End Do
End Do

        !!!  THE WIDE BIN UNFOLDING RESPONSE  !!!

! The nested do loop which follows calculates a wide bin response function
! which is used for unfolding.  This is a requirement for the code because
! only a square matrix can be inverted, and inversion of the response
! matrix is necessary for use of the Subroutine Jacobi.  In essence, the
! narrow bin response function (an i x j array, non-square) is converted
! by use of the Sum intrinsic function to a wide bin response function
! (an i x k array, square).

Do i=1,iMax
   Do k=1,nw
      RwUnfoldExact(i,k) = Sum(RnUnfold(i,jMax(k-1)+1:jMax(k)))
   End Do
End Do

        !!!  GENERATION OF THE SYSTEMATIC RESPONSE ERROR  !!!

! The following case statement generates a systematic error to be added to
! the exact response function.  The user can generate the random error, or
! he can let the machine do it.  If the user generates the error, he can
! choose between an uniform distribution or a gaussian distribution.  In
! either case, this error generation is based on the random seed input by
! the user.  If the machine generates the error, the same two options are
! available (uniform or gaussian).  The only difference comes with the seed.
! In this case, it is based on a call of the system clock.

Allocate(R1A(iMax), R2A(iMax,12), G1A(iMax))

If (SysErrorPercentRespns > 0) Then
  Select Case(ResponseSysErrorChoice)
      Case(1)
          Select Case(ResSysErrorFormat)
             Case(1)
                 Call Random_Seed(Put = SeedInput2)
                 Call Random_Number(R1A)
                 Do i=1,iMax
                     RwUnfoldSysError(i,:) = SysErrorPercentRespns*((2*R1A)-1)*  &
```

```
                      RwUnfoldExact(i,:)/100
                  End Do
              Case(2)
                  Call Random_Seed(Put = SeedInput2)
                  Call Random_Number(R2A)
                  G1A = Sum(R2A,2) - 6
                  Do i=1,iMax
                         RwUnfoldSysError(i,:) = SysErrorPercentRespns*G1A*  &
                         RwUnfoldExact(i,:)/100
                  End Do
           End Select
       Case(2)
           Select Case(ResSysErrorFormat)
              Case(1)
                  Call System_Clock(Count2)
                  Seed2 = Count2
                  Call Random_Seed(Put = Seed2)
                  Call Random_Number(R1A)
                  Do i=1,iMax
                      RwUnfoldSysError(i,:) = SysErrorPercentRespns*((2*R1A)-1)*  &
                      RwUnfoldExact(i,:)/100
                  End Do
              Case(2)
                  Call System_Clock(Count2)
                  Seed2 = Count2
                  Call Random_Seed(Put = Seed2)
                  Call Random_Number(R2A)
                  G1A = Sum(R2A,2) - 6
                  Do i=1,iMax
                      RwUnfoldSysError(i,:) = SysErrorPercentRespns*G1A*  &
                      RwUnfoldExact(i,:)/100
                  End Do
           End Select
  End Select
Else
  RwUnfoldSysError = 0
End If

Deallocate(R1A, R2A, G1A)

        !!!  GENERATION OF THE ENERGY DEPENDENT RESPONSE ERROR  !!!

! The following case statement generates an energy dependent error to be added
! to the sum of the exact wide bin response function and the systematic error
! for the wide bin unfolding response function.  The user can generate the
! random error, or he can let the machine do it.  If the user generates the
! error, he can choose between an uniform distribution or a gaussian
! distribution.  In either case, this error generation is based on the random
! seed input by the user.  If the machine generates the error, the same two
! options are available (uniform or gaussian).  The only difference comes with
! the seed.  In this case, it is based on a machine call of the system clock.

Allocate(R2(iMax,nw), G2(iMax,nw), R3(imax,nw,12))

If (BinErrorPercentRespns > 0) Then
  Select Case(ResponseBinErrorChoice)
     Case(1)
        Select Case(ResBinErrorFormat)
           Case(1)
              Call Random_Seed(Put = SeedInput3)
              Call Random_Number(R2)
              RwUnfoldBinError = BinErrorPercentRespns*((2*R2)-1)*     &
              (RwUnfoldExact+RwUnfoldSysError)/100
           Case(2)
              Call Random_Seed(Put = SeedInput3)
              Call Random_Number(R3)
              G2 = Sum(R3,3) - 6
              RwUnfoldBinError = BinErrorPercentRespns*G2*     &
              (RwUnfoldExact+RwUnfoldSysError)/100
        End Select
     Case(2)
        Select Case(ResBinErrorFormat)
           Case(1)
              Call System_Clock(Count3)
              Seed3 = Count3
              Call Random_Seed(Put = Seed3)
              Call Random_Number(R2)
              RwUnfoldBinError = BinErrorPercentRespns*((2*R2)-1)*  &
              (RwUnfoldExact+RwUnfoldSysError)/100
           Case(2)
              Call System_Clock(Count3)
              Seed3 = Count3
              Call Random_Seed(Put = Seed3)
              Call Random_Number(R3)
              G2 = Sum(R3,3) - 6
              RwUnfoldBinError = BinErrorPercentRespns*G2*  &
              (RwUnfoldExact+RwUnfoldSysError)/100
        End Select
  End Select
Else
```

```
   RwUnfoldBinError = 0
End If

Deallocate(R2, G2, R3)

        !!!  THE WIDE BIN CALIBRATED UNFOLDING RESPONSE  !!!

! The nested do loop below calculates a simulated wide bin calibrated
! response function.  It is composed of an exact component and an error
! component.  The error component simulates mistakes made in the
! calibration of the instrument.  This calibrated response function is the
! function used for unfolding.

RwUnfoldCalib = RwUnfoldExact + RwUnfoldSysError + RwUnfoldBinError

        !!!  THE NARROW BIN FOLDING FLUXES  !!!

! The looping structure coded below calculates the narrow bin known fluxes
! which simulate the environment within which the instrument is placed.
! The form of the flux is input by the user.  These are the fluxes which
! are folded with the narrow bin response functions which, in turn,
! produce the exact signals.

Do j=1,jTotal
   Select Case (sChoice)
      Case(1)
         nFlux(j) = Maxwellian(En(j), temperature)
      Case(2)
         nFlux(j) = ExponentialD(En(j))
      Case(3)
         nFlux(j) = LinearD(En(j))
      Case(4)
         nFlux(j) = ConExponD(En(j))
      Case(5)
         nFlux(j) = ConLinD(En(j))
      Case(6)
         nFlux(j) = Heaviside(En(j))
      Case(7)
         nFlux(j) = ActualFlux(En(j), power)
   End Select
End Do

        !!!  THE WIDE BIN KNOWN FLUX  !!!

! The do loop below calculates a wide bin known flux.  This CANNOT be
! used for folding. Its sole purpose is as a comparison with the wide
! bin unfolded flux.  The wide bin known flux is simply an average of the
! narrow bin fluxes contained within that wide bin.

Do k=1,nw
   wFlux(k) = Sum(nFlux(jMax(k-1)+1:jMax(k)))/nn(k)
End Do

        !!!  THE EXACT INSTRUMENT SIGNALS  !!!

! The statement below calculates the ideal signal measured in each channel
! of the instrument.  It is important to note that this must be done as the
! instrument does it, so the signals are calculated by narrow bin fluxes
! and narrow bin response functions.  In other words, the flux is folded
! with the response function to generate the signals.

Allocate(yExact(iMax), yDelta(iMax), yMeasured(iMax))
Allocate(R1(iMax), R2(iMax,12), G1(iMax))

yExact = MatMul(RnFold,nFlux)

        !!!  GENERATION OF THE COUNTING ERROR FOR THE SIGNALS  !!!

! The following case statement generates a counting error to be added to
! the ideal instrument signal.  The user can generate the random error, or
! he can let the machine do it.  If the user generates the counting error,
! he must input the seed.  If the machine generates the error, the seed is
! based on a call of the system clock.

yMax = MaxVal(yExact)

If (ErrorPercentSignal > 0) Then
  Select Case(SignalErrorChoice)
     Case(1)
        Call Random_Seed(Put = SeedInput1)
        Call Random_Number(R2)
        G1 = Sum(R2,2) - 6
        yDelta = (ErrorPercentSignal/100)*G1*((yMax/yExact)**0.5)*yExact
     Case(2)
        Call System_Clock(Count1)
        Seed1 = Count1
        Call Random_Seed(Put = Seed1)
        Call Random_Number(R2)
        G1 = Sum(R2,2) - 6
        yDelta = (ErrorPercentSignal/100)*G1*((yMax/yExact)**0.5)*yExact
  End Select
 Else
  yDelta = 0
 End If

 Deallocate(R1, R2, G1)

         !!!  THE MEASURED SIGNALS  !!!

 ! The statement below calculates the signals actually measured by the
 ! instrument.  These measured signals have two components; the exact
 ! signal resulting from the flux, and the error introduced by poor
 ! detector performance.

 yMeasured = yExact + yDelta

         !!!  SCREEN OUTPUT FOR SIGNALS AND RESPONSES  !!!

 ! The following print section of code ensures the user can judge the
 ! performance of the code by tracking various signal and response elements
 ! used in the calculation of the different fluxes.  Basically, it shows as
 ! output the difference between ideal and real measurements and responses.

Print*, "      Channel Exact Signal  Measured Signal    Ideal Rspns   Calib'd Rspns"
Do i=1,iMax
  Print 20, i, yExact(i), yMeasured(i), RwUnfoldExact(i,i), RwUnfoldCalib(i,i)
  20 Format('       ', I3: '   ',F12.8:'   ',F12.8:'       ',F13.8:'   ',F13.8)
End Do
Print*

         !!!  CALL JACOBI, CALCULATE THE UNFOLDED FLUXES  !!!

! The section of code below is the core of the program.  It is the call for
! the subroutine Jacobi, the subprogram which solves for the unfolded flux
! by using the Jacobi iterative technique.  The wide bin calibrated
! response function and the measured signals are passed to Jacobi.  The
! subroutine then returns the unfolded flux.

Allocate(FluxUnfolded(nw))

If (nw==iMax) Then
  Call Jacobi(FluxUnfolded,RwUnfoldCalib,yMeasured)
Else
  Print*, "Nonsquare matrix, not yet supported!"
  STOP
```

```
End If

        !!! CALCULATIONS FOR COMPARISONS !!!

! The section below defines some basic quantities of interest. The key
! point is the percent difference between the exact and the unfolded
! fluxes. This is, in essence, the whole reason for writing this code.
! NOTE, NOTE, NOTE : if negative fluxes are unfolded, the following
! quantities will not be correct!!

Allocate(DeltaWideFlux(iMax), FluxErrorPcnt(iMax))
Allocate(DeltaSig(iMax), DeltaRes(iMax,nw))

DeltaWideFlux = FluxUnfolded - wFlux
FluxErrorPcnt = (DeltaWideFlux/wFlux)*100
DeltaSig = yExact - yMeasured
DeltaRes = RwUnfoldExact - RwUnfoldCalib

        !!! SCREEN OUTPUT FOR FLUXES AND DIFFERENCES !!!

! The following print statement shows the critical components of the
! output. The key values to note are the exact flux, the unfolded flux,
! and the percent difference between the two. If this percent difference
! is positive, it means the unfolded flux is larger than the exact flux.
! The delta signal and the delta response values are given so that the
! user can note the performance of the error generating functions.

Print*, "   Wide Bin  Exact Flux    Unfolded Flux  Flux %Err  DSig      DRes"

Do i=1,nw
  Print 10, i, wFlux(i), FluxUnfolded(i), FluxErrorPcnt(i), DeltaSig(i), &
  DeltaRes(i,i)
  10 Format('      ',I3:'  ',F11.8:'  ',F11.8:'   ',F8.2:'    ',F8.6:&
  '  ',F8.6)
End Do

! The section of code below writes important output arrays to data
! files. These files can be plotted in GeneralizedBarChart functions
! found in Mathematica, v2.2.

If (WriteChoice .EQ. 1) Then
   Open (Unit = 1, File = 'C:\wnmath22\fluxunfd.dat')
   Write (1,*) FluxUnfolded
   Close (1)
```

```
    Open (Unit = 25, File = 'C:\wnmath22\dEw.dat')
    Write (25,*) dEw
    Close (25)

    Open (Unit = 21, File = 'C:\wnmath22\Ebw.dat')
    Write (21,*) Ebw
    Close (21)

    Open (Unit = 22, File = 'C:\wnmath22\Ebc.dat')
    Write (22,*) Ebc
    Close (22)

    Open (Unit = 2, File = 'C:\wnmath22\fluxknwn.dat')
    Write (2,*) wFlux
    Close (2)
End If

End Program InstrumentUnfold

! The main program is finished.  What follows is the section of code which
! defines the ten functions and one subroutine used by the main program.

! The following function calculates the energy value for each narrow bin if the
! the electron distribution is Maxwellian in nature.

Real(8) Function Maxwellian(e,tau)

    Implicit None
    Real(8) e                           ! energy of particles
    Real(8) tau                         ! fundamental temperature

    Maxwellian = 1.12837917*Sqrt(e/tau)*Exp(-e/tau)/tau

End Function Maxwellian

! The following function calculates the energy value for each narrow bin if the
! the electron distribution experiences an exponential decrease with a maximum
! number of electrons found at 1 MeV.

Real(8) Function ExponentialD(e)

    Implicit None
    Real(8) e                           ! energy of the particle
    Real(8), Parameter:: PhiInit = 1.0  ! initial flux
    Real(8), Parameter:: eInit = 1.0    ! scaling value
```

```
   If (e <= 1.0) Then
      ExponentialD = PhiInit
   Else
      ExponentialD = PhiInit*Exp(eInit/e)/2.71828182846
   End If

End Function ExponentialD

! The following function calculates the energy value for each narrow bin if the
! the electron distribution experiences a linear decrease with a maximum
! number of electrons found at 1 MeV.

Real(8) Function LinearD(e)

   Implicit None
   Real(8) e                              ! energy of the particle
   Real(8), Parameter:: PhiInit = 1.0     ! initial flux

   If (e <= 1.0) Then
      LinearD = PhiInit
   Else
      LinearD = PhiInit*(1 - (e-1)/10)    ! Hardwired parameters here
   End If

End Function LinearD

! The following function calculates the energy value for each narrow bin if the
! the electron distribution experiences an exponential decrease after maintaining
! a constant value from 1 to 3 MeVs.  The maximum number of electrons is the
! constant value found from 1 to 3 MeVs.

Real(8) Function ConExponD(e)

   Implicit None
   Real(8) e                              ! energy of the particle
   Real(8), Parameter:: PhiInit = 1       ! initial flux
   Real(8), Parameter:: eInit = 3         ! scaling value

   If (e <= 3.0) Then
      ConExponD = PhiInit
   Else
      ConExponD = PhiInit*Exp(eInit/e)/2.71828182846
   End If
End Function ConExponD

! The following function calculates the energy value for each narrow bin if the
! the electron distribution experiences a linear decrease after maintaining
! a constant value from 1 to 3 MeVs. The maximum number of electrons is the
! constant value found from 1 to 3 MeVs.

Real(8) Function ConLinD(e)

   Implicit None
   Real(8) e                              ! energy of the particle
   Real(8), Parameter:: PhiInit = 1.0     ! initial flux
   Real(8), Parameter:: eInit = 3.0       ! scaling value

   If (e <= 3.0) Then
      ConLinD = PhiInit
   Else
      ConLinD = PhiInit*(1-(e-3)/8)       ! Hardwired parameters here
   End If

End Function ConLinD

! The following function calculates the energy value for each narrow bin if the
! the electron distribution takes the form of a Heaviside function. The number
! of electrons remain constant from 1 to 3 MeVs, and then experience a factor
! of 100 decrease for the remainder of the interval of interest.

Real(8) Function Heaviside(e)

   Implicit None
   Real(8) e                              ! energy of the particle
   Real(8), Parameter:: PhiInit = 1.0     ! initial flux

   If (e <= 3.0) Then
      Heaviside = PhiInit
   Else
      Heaviside = PhiInit/100
   End If

End Function Heaviside

! The following function calculates the energy value for each narrow bin based
! on a flux form input by the sponsor at Phillips Lab. This rapidly decaying
! form for the flux is based upon actual measurements made by the HEEF.
```

```
Real(8) Function ActualFlux(e,p)

   Implicit None
   Real(8) e                          ! energy of the particle
   Real(8) p                          ! value for the exponent
   Real(8), Parameter:: PhiInit = 1.0 ! initial flux

   If (e <= 1.0) Then
      ActualFlux = PhiInit
   Else
      ActualFlux = PhiInit*e**(-p)
   End If

End Function ActualFlux

! The following function defines an idealized response for each narrow bin of the
! instrument.  Although not physically realistic, it is extremely useful as a
! tool for which to check the values of the calculated data.  It can serve as
! a crude approximation for the first calibration of the HEEF.

Real(8) Function ResponseA(i,e,wbeb)

   Implicit None
   Integer i
   Real(8) e
   Real(8), Dimension(0:) :: wbeb
   Real(8), Dimension(10) :: Rpeak
   Real(8) eMin, eMax

   Rpeak = (/0.00010397, 0.00046705, 0.00075258, 0.00090856, 0.00119878, &
      0.00134863, 0.00347621, 0.00427466, 0.00930346, 0.00984686/)

   eMin = wbeb(i - 1)
   eMax = wbeb(i)
   If (e<eMin) then
      ResponseA = 0.0
   Else if (e<=eMax) then
      ResponseA = Rpeak(i)
   Else
      ResponseA = 0.0
   End if

End Function ResponseA

! The following function defines a heaviside response for the instrument.
! This serves as a crude approximation for the second calibration
! performed on the HEEF.

Real(8) Function ResponseB(i,e,wbeb)

   Implicit None
   Integer i
   Real(8) e
   Real(8), Dimension(0:) :: wbeb
   Real(8), Dimension(10) :: Rpeak
   Real(8) eMin

   Rpeak = (/0.00019227, 0.00030235, 0.00078752, 0.00108250, 0.00147380, &
      0.00224757, 0.00396249, 0.00695279, 0.00957538, 0.00818810/)

   eMin = wbeb(i-1)
   If (e<eMin) then
      ResponseB = 0
   Else
      ResponseB = Rpeak(i)
   End if

End Function ResponseB

! The following function defines a guassian form of the response for
! the instrument. This serves as a more-refined approximation for the
! first calibration performed on the HEEF.

Real(8) Function ResponseC(i,e,wbeb)

   Implicit None
   Integer i
   Real(8) e
   Real(8), Dimension(0:) :: wbeb
   Real(8), Dimension(10) :: Rpeak
   Real(8) eMin, eMax, eAve
   Real(8),Parameter:: sigma = 0.1

   Rpeak = (/0.00010397, 0.00046705, 0.00075258, 0.00090856, 0.00119878, &
        0.00134863, 0.00347621, 0.00427466, 0.00930346, 0.00984686/)

   eMin = wbeb(i-1)
   eMax = wbeb(i)
   eAve = (eMin + eMax)/2
   If (e<eAve) Then
```

```
    ResponseC = Exp(-(eAve-e)*(eAve-e)/(2*sigma*sigma))*Rpeak(i)
  Else
    ResponseC= Exp(-(e-eAve)*(e-eAve)/(2*sigma*sigma))*Rpeak(i)
  End If

End Function ResponseC

! The following function provides a more-refined estimate of the
! second calibrated instrument response.

Real(8) Function ResponseD(i,e,wbeb)

  Implicit None
  Integer i
  Real(8) e
  Real(8), Dimension(0:) :: wbeb
  Real(8), Dimension(10) :: Rpeak
  Real(8) eMin, eMax, eAve
  Real(8),Parameter:: sigma = 0.1

  Rpeak = (/0.00019227, 0.00030235, 0.00078752, 0.00108250, 0.00147380, &
      0.00224757, 0.00396249, 0.00695279, 0.00957538, 0.00818810/)

  eMin = wbeb(i-1)
  eMax = wbeb(i)
  eAve = (eMin + eMax)/2
  If (e<eAve) then
    ResponseD = Exp(-(eAve-e)*(eAve-e)/(2*sigma*sigma))*Rpeak(i)
  Else
    ResponseD = Rpeak(i)
  End if

End Function ResponseD

! Below lies the code which calculates the response function for
! the first calibration of the HEEF.

Real(8) Function HEEFresOne(i,j,e,jTotalpass)

  Implicit None

  Integer i
  Integer j
  Integer jTotalpass
```

```
Real(8) e
Real(8), Dimension(10) :: Ep
Real(8), Dimension(10) :: Rmax
Real(8), Dimension(10) :: Sigma
Real(8), Dimension(10) :: DeltaE
Real(8) GeoFactor
Real(8) RelativeRes

! This section of the code calculates the geometric factor.  The
! analytical functions used are slightly altered from those listed
! in the calibration report.  This change is detailed in chapter
! three of the thesis.

If (j<=jTotalpass) Then
   If (e<=1.75) Then
     GeoFactor = exp((-5.829*e + 21.452)*e - 14.985)
   Else if (e<=2.8) Then
     GeoFactor = exp((-0.378*e + 2.553)*e + 1.373)
   Else
     GeoFactor = 700*(1-1.69/(e+0.2))**1.2
   End If
   Else
     STOP "In the Function HEEFresOne the j index is not correct"
End If

! The arrays which follow are:
!The energy values in each wide bin where the response
!    is peaked
!The peak response in each wide bin
!The values for sigma (taken from the calibration report)
!The values for delta E (taken from the calibration report)

Ep = (/1.30, 1.82, 2.35, 2.80, 3.30, &
     3.80, 4.55, 5.55, 7.08, 9.05/)

Rmax = (/0.919, 0.914, 0.925, 0.896, 0.886, &
     0.905, 0.997, 0.997, 1.000, 1.000/)

Sigma = (/0.234, 0.234, 0.234, 0.221, 0.234, &
     0.221, 0.293, 0.340, 0.357, 0.425/)

DeltaE = (/0.00, 0.00, 0.00, 0.00, 0.00, &
     0.00, 0.15, 0.15, 0.58, 0.50/)

! This portion of the code calculates the relative response for
```

```
   ! the first calibration.  There were obvious errors in the
   ! technical report, so the analytical expressions used are
   ! different from the ones given in the paper.

   If (i<= 6) Then
      RelativeRes = Rmax(i)*exp(-(e-Ep(i))**2/(2*Sigma(i)**2))
      Else if (i<=10) Then
         If (e < Ep(i)-DeltaE(i)) Then
            RelativeRes = Rmax(i)*exp(-(e-Ep(i)+DeltaE(i))**2/  &
                        (2*Sigma(i)**2))
         Else if (e > Ep(i)+DeltaE(i)) Then
            RelativeRes = Rmax(i)*exp(-(e-Ep(i)-DeltaE(i))**2/  &
                        (2*Sigma(i)**2))
         Else
               RelativeRes = Rmax(i)
         End If
      Else
         RelativeRes = (0.1316*e**3) - (2.8149*e**2) + (14.991*e) + 0.659
   End If

   ! This section of code calculates the absolute response as
   ! determined by the first calibration of the HEEF.  There is a
   ! scaling factor in this relation which scales the values
   ! calculated in the code with those shown in the plot in the
   ! technical report.  Why this difference in response calculations
   ! exists is not known, but it may be a mistake in the report.

   HEEFresOne = (GeoFactor * RelativeRes)/(1E5)

End Function HEEFresOne

! Below lies the code which calculates the response function for
! the second calibration of the HEEF.

Real(8) Function HEEFresTwo(i,e)

   Implicit None

   Integer, Intent(IN):: i
   Real(8), Intent(IN):: e
   Real(8) x, fd
   Integer, Parameter :: iMax = 10

   Real(8), Dimension(iMax) :: Epeak
   Real(8), Dimension(iMax) :: Rpeak
   Real(8), Dimension(iMax) :: Etail
   Real(8), Dimension(iMax) :: Sigma
   Real(8), Dimension(iMax) :: tailFrac

   ! The values for the arrays which follow are taken from table 2
   ! in chapter three of the thesis. No analytical expressions for
   ! the absolute response derived from the partial re-calibration were
   ! listed, so we made our own. Chapter three explains the rationale
   ! used to construct these responses.

   Epeak = (/1.35, 1.70, 2.25, 2.80, 3.30,  &
        3.80, 4.55, 5.55, 7.08, 9.05/)

   Rpeak = (/6.0E-04, 1.0E-03, 1.3E-03, 2.5E-03, 3.2E-03, &
        3.7E-03, 4.6E-03, 5.1E-03, 5.6E-03, 6.0E-03/)

   Etail = (/2.02, 2.73, 3.38, 4.20, 4.95,  &
        5.70, 6.83, 8.33, 10.62, 13.58/)

   Sigma = (/0.170, 0.161, 0.340, 0.221, 0.234,  &
        0.221, 0.293, 0.340, 0.357, 0.425/)

   tailFrac = (/0.80, 0.52, 0.50, 0.50, 0.50,  &
        0.50, 0.50, 0.50, 0.50, 0.50/)

   If (i<1 .or. i>iMax) STOP "counting index for i is not correct!"

   If (e < Epeak(i)) Then
      HEEFresTwo = Exp(-(Epeak(i)-e)*(Epeak(i)-e)/(2*sigma(i)*sigma(i)))
   Else if (e < Etail(i)) Then
      x = (e - Epeak(i))/(Etail(i) - Epeak(i))     ! interpolate between Epk and Etail
      fd = 1 - tailFrac(i)                         ! fraction decrease from peak in tail
      HEEFresTwo = 1 + x * x * fd * (-3 + 2*x)
   Else
      HEEFresTwo = tailFrac(i)
   End If

   HEEFresTwo = HEEFresTwo * Rpeak(i)

End Function HEEFresTwo

! This subroutine solves the linear system Ax = b in an iterative
! fashion by utilizing the Jacobi technique.

Subroutine Jacobi(x,A,b)
```

```
Real(8), Intent(Out):: x(:)
Real(8), Intent(In):: A(:,:), b(:)
Real(8), Allocatable:: T(:,:), c(:), Dinv(:,:), xold(:), r(:)

Integer i,n

! This constant is the relative error tolerance used to terminate
! the iterations when the difference between successive flux vectors
! is small.

Real(8), Parameter:: tol=1.D-8

n = size(b,1)

! A final check to ensure all required parameters contain the
! correct number of elements.

If (Size(A,1)/=n .or. Size(A,2)/=n .or. Size(x,1)/=n) Then
   STOP "Incompatible argument dimensioning in Jacobi."
End If

Allocate(T(n,n), Dinv(n,n), c(n), xold(n), r(n))

! This section of the code assigns values to matrices used in the
! Jacobi iterative scheme.  See chapter two of the thesis.

Dinv = 0
T =  A

Do i=1,n
   If (A(i,i)==0) STOP "Zero diagonal element in Jacobi"
   Dinv(i,i) = 1/A(i,i)
   T(i,i) = 0
End Do

T = -MatMul(Dinv,T)
c = MatMul(Dinv,b)

x = c                          ! Starting guess

! This is the loop which actually computes the values of the
! unfolded fluxes by using the Jacobi method.  See chapter two
! for a detailed explanation of this technique.
```

```
 Do
    xold = x
    x = MatMul(T,x) + c
    r = MatMul(A,x)-b
    If (All(abs(r)<=tol*Abs(b))) Return
 End Do
 Deallocate(r, xold, c, Dinv, T)
End Subroutine Jacobi
```

Appendix B: A Case Study for Comparison

This appendix contains a notebook shell for checking the Fortran 90 results in Mathematica, v2.2. This serves as an excellent way to compare exact and unfolded fluxes with the output generated by the code. This case study is for a flux modeled by

$$\varphi(E) = E^{-4} \tag{B-1}$$

and a heaviside response function. There is no error added for the calibration or for the measurements, so the difference between the exact flux and the unfolded flux should be zero.

The first step is selecting a flux format.

```
f[e_] = 1/e^4     [hit insert]
```

The second step is generating a set of instrument signals. The measurements are alculated by integrating the flux. Remember to account for the width of the wide bins.

```
y = Join[Table[Integrate[f[e],{e,i,12}],{i,1,9}],
      {Integrate[f[e],{e,10,12}]}]                    [hit insert]
```

```
N[y]                  [hit insert]
```

This table lists the values for the response matrix and displays them in matrix form. The values were known from calculations done in the Fortran 90 code, so this table just reproduces them.

```
r = Table[
  Join[Table[0,{j,1,i-1}],
        Table[1,{j,i,9}],{2}],

   {i,1,10}];                [hit insert]

r // MatrixForm              [hit insert]
```

This step performs the calculation for the exaxt flux.

```
phiExact = Join[Table[Integrate[f[e],{e,i,i+1}],

        {i,1,9}],{Integrate[f[e],{e,10,12}]/2}]       [hit insert]

phiExact // N            [hit insert]
```

The unfolded flux is calculated by linear solving r and y.

```
phiUnfold = LinearSolve[r,y]             [hit insert]

phiUnfold // N            [hit insert]
```

This step calculates the difference between the two fluxes. Since no calibration or measurement error was added, the difference should be zero. This is a good tool for use in comparing results with the Fortran 90 code.

```
phiUnfold - phiExact          [hit insert]
```

Bibliography

1. Ball Space Systems Division. CRRES System Description Handbook. Prepared for the National Aeronautics and Space Administration, George C. Marshall Space Flight Center, in accordance with Data Requirement SE-11 of Contract No. NAS8-34025, no date.

2. Burden, Richard L. and Faires, J. Douglas. Numerical Analysis, Fifth Edition. Boston, MA: PWS Publishing Company, 1993.

3. Dichter, Bronislaw K. and Hanser, Frederick A. Development and Use of Data Analysis Procedures for the CRRES Payloads AFGL-70102/Dosimeter and AFGL-701-4/Fluxmeter and Application of the Data Analysis Results to Improve the Static and Dynamic Models of the Earth's Radiation Belts. Document GL-TR-89-0284, Scientific Report No. 2. Waltham, MA: Panametrics, Inc., October, 1989.

4. Dichter, Bronislaw K. and Hanser, Frederick A. Development and Use of Data Analysis Procedures for the CRRES Payloads AFGL-70102/Dosimeter and AFGL-701-4/Fluxmeter and Application of the Data Analysis Results to Improve the Static and Dynamic Models of the Earth's Radiation Belts. Document PL-TR-91-2186, Scientific Report No. 3. Waltham, MA: Panametrics, Inc., July, 1991.

5. Frederickson, A. Robert. Geophysics Directorate, USAF Phillips Laboratory, Hanscom AFB MA. Telephone interview. 3 September 1996

6. Hanser, Frederick A. Analyze Data from CRRES Payloads AFGL-710-2/Dosimeter and AFGL-701-4/HEEF. Document PL-TR-93-, Scientific Report No. 1. Waltham, MA: Panametrics, Inc., February, 1993.

7. M. Johnson, and J. Kierein, "Combined Release and Radiation Effects Satellite (CRRES): Spacecraft and Mission," Journal of Spacecraft and Rockets, Volume 29, No. 4, pp. 556-563, 1992

8. Kivelson, Margaret G. and Russell, Christopher T. Introduction to Space Physics. New York, NY: Cambridge University Press, 1995.

9. Liebowitz, Ruth P. A History of the Space Radiation Effects (SPACERAD) Program for the Joint USAF/NASA CRRES Mission, Part I: From Origins Through the Launch, 1981-1990. Document PL-TR-92-2071, Special Report No. 269. Hanscom AFB, MA: Phillips Laboratory, Directorate of Geophysics, Air Force Systems Command, March 1992.

10. K. Mathews, "Spectra of Cooling Blackbody Emitters as Basis Functions for Unfolding Integral Spectrum Measurements," Journal of Radiation Effects, Research and Engineering, Volume 12, No.2, pp. 17-23, 1994.

11. K. Mathews, and D. Miller, “Smoothing Techniques for Iterative Unfolding of Diagnostic Spectra,” Journal of Radiation Effects, Research and Engineering, Volume 12, No. 2, pp. 38-60, 1994.

12. D. Miller and K. Mathews, “Application of Cross Validation Method to Iterative Unfolding of Diagnostic Spectra,” Journal of Radiation Effects, Research and Engineering, Volume 12, No. 2, pp. 24-37, 1994.

13. Parks, George K. Physics of Space Plasmas. Redwood City, CA: Addison-Wesley Publishing Company, 1991.

14. A. Vampola, “Combined Release and Radiation Effects Satellite,” Journal of Spacecraft and Rockets, Volume 29, No. 4, p. 555, 1992.

15. Wing, G. Milton. A Primer on Integral Equations of the First Kind. Baltimore, MD: Port City Press, 1991.

Vita

Capt Brian D. McKellar was born on 21 September 1967 in Freeland, Michigan. He graduated from Freeland High School in 1986 and entered undergraduate studies at the Michigan State University in East Lansing, Michigan. He graduated with a Bachelor of Science degree with a major in astrophysics in September 1990. He received his commission on 3 August 1990 upon graduation from the Reserve Officer Training Corps.

His first assignment was at the Florida State University in Tallahassee, Florida as a student in the Basic Meteorology Program. He received a legal certification in meteorology on 25 April 1992. His next assignment was at Mountain Home AFB, Idaho as a wing weather officer for the 366th Composite Wing. His third assignment was at Yongsan AIN, Republic of Korea as a staff weather officer for the 17th Aviation Brigade. In May 1995, he entered the Department of Engineering Physics in the School of Engineering at the Air Force Institute of Technology.

Permanent Address: 8227 Midland Rd.
Freeland, MI 48623

# REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 1996	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE Unfolding The High Energy Electron Flux From CRRES Fluxmeter Measurements	**5. FUNDING NUMBERS**
6. AUTHOR(S) Brian D. McKellar, Capt, USAF	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology 2750 P Street WPAFB OH 45433-6583	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GAP/ENP/96D-08
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Dr. Arthur Robb Frederickson Phillips Laboratory, Geophysics Directorate Space Physics Division 29 Randolph Rd. Hanscom AFB MA 01731-3010	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT *(Maximum 200 words)*

The Combined Release and Radiation Effects Satellite (CRRES) was launched on 25 July 1990 to collect measurements in the earth's radiation belts. One instrument, the High Energy Electron Fluxmeter (HEEF), measured the flux of electrons in 10 channels with energies between 1 MeV and 10 MeV. The channel sensitivities, $R_i(E)$, have been calibrated and partially re-calibrated. We explore the errors introduced in unfolding the electron flux spectrum from the channel measurements and the propagation and growth of calibration and measurement errors. Using numerical experimentation, we fold the responses with known spectra to obtain simulated measurements, add random measurement and calibration errors, and unfold the spectra as 10-bin histograms which are compared with histograms of the original spectra. We observe that the shape (of the response functions) is the major factor in the growth of error in unfolding and in determining which type of error dominates the unfolding process. We conclude that successful unfolding of the electron flux is critically dependent upon the shape of the response functions. The re-calibration of the HEEF must be accurately completed if reliable unfolds of the high energy electron flux are to be obtained.

14. SUBJECT TERMS CRRES, High Energy Electron Fluxmeter, Electrons, Magnetosphere, Radiation Belts, Unfolding	**15. NUMBER OF PAGES** 131
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C - Contract
G - Grant
PE - Program Element
PR - Project
TA - Task
WU - Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE - See authorities.
NASA - See Handbook NHB 2200.2.
NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.
DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA - Leave blank.
NTIS - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

★U.S.GPO:1993-0-336-043